

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

_____ Сергій СТИПЕНКО

«__» _____ 20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Комп'ютерні системи та мережі»

спеціальності 123 «Комп'ютерна інженерія»

**на тему: «Система автоматизованого інформування іноземних
студентів»**

Виконав:

студент IV курсу, групи ІО-62

Якубовський Андрій Юрійович

Керівник:

Доцент, кандидат технічних наук,

Павлов Валерій Георгійович

Консультант з нормоконтролю:

Професор, доктор технічних наук

Сімоненко Валерій Павлович

Рецензент:

Професор, доктор технічних наук,

Корнієнко Богдан Ярославович

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій СТИПЕНКО

«__» _____ 20__ р.

ЗАВДАННЯ
на дипломний проєкт студенту
Якубовському Андрію Юрійовичу

1. Тема проєкту «Система автоматизованого інформування іноземних студентів», керівник проєкту Павлов Валерій Георгійович, доцент, к.т.н., затверджені наказом по університету від «07» травня 2020 р. № 1081-с
2. Термін подання студентом проєкту 06 червня 2020 р.
3. Вихідні дані до проєкту: технічне завдання, науково-технічна література
4. Зміст пояснювальної записки: порівняльний аналіз існуючих програмних рішень, вибір засобів реалізації та опис отриманої системи
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) :
 1. Діаграма класів проєкту – плакат;
 2. Алгоритм сортування вставками – плакат;
 3. Загальна структура системи – плакат.

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сімоненко В.П., професор		

7. Дата видачі завдання 01 вересня 2019 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Затвердження теми роботи	01.09.2019-22.12.2019	
2	Вивчення та аналіз завдання	23.12.2019-22.03.2020	
3	Проведення порівняльного аналізу з існуючими аналогами	23.03.2020-01.04.2020	
4	Проектування та розробка бази даних програмного продукту	02.04.2020-10.04.2020	
5	Розробка серверної частини	11.04.2020-20.04.2020	
6	Розробка клієнтської частини	21.04.2020-30.04.2020	
7	Оформлення пояснювальної записки	01.05.2020-23.05.2020	
8	Передзахист	24.05.2020-26.05.2020	
9	Захист	15.06.2020-20.06.2020	

Студент

Андрій ЯКУБОВСЬКИЙ

Керівник

Валерій ПАВЛОВ

АНОТАЦІЯ

Даний дипломний проєкт присвячений розробці автоматизованої системи інформування іноземних студентів.

Дана інформаційна система покликана задля того, щоб допомогти іноземним студентам, оскільки саме ця група людей, частіше за все, обмежена у своїх можливостях вчасно отримувати важливу інформацію через мовний бар'єр, період адаптації в новій для них країні або звичайну невпевненість. Тому було прийнято рішення створити автоматизовану систему інформування, задля вирішення проблеми вчасного інформування іноземних студентів і оптимізації робочого процесу викладачів.

ANNOTATION

This thesis is devoted to the development of an automated system for informing foreign students.

This information system is designed to help foreign students, as this group of people is often limited in their ability to receive important information in a timely manner due to the language barrier, the period of adaptation in a new country or the usual unconfident. Therefore, it was decided to create an automated information system to solve the problem of timely informing foreign students and optimizing the work process of teachers.

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проєкт	2	
2	A4	ДП 4671. 00.000 ВП	Відомість проєкту	1	
3	A4	ДП 4671. 01.000 ТЗ	Технічне завдання	4	
4	A4	ДП 4671. 02.000 ПЗ	Пояснювальна записка	62	
5	A4	ДП 4671. 03.000 Д1	Функціональна схема діаграми класів проєкту	1	
6	A4	ДП 4671. 04.000 Д2	Принципова схема алгоритму сортування вставками	1	
7	A4	ДП 4671. 05.000 Д3	Структурна схема системи	1	

				ДП 4671 00.000 ВП		
	ПІБ	Підп.	Дата	Відомість дипломного проєкту	Лист	Листів
Розробник	Якубовський А.Ю.				1	1
Керівник	Павлов В.Г.				КПІ ім. Ігоря Сікорського Каф. ОТ Гр. ІО-62	
Консульт.						
Н/контр.	Сімоненко В.П.					
Зав.каф.	Стіренко С.Г.					

Технічне завдання
до дипломного проєкту
на тему: «Система автоматизованого інформування
іноземних студентів»

Київ – 2020 року

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ.....	2
3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ	2
4. ДЖЕРЕЛА РОЗРОБКИ	2
5. ТЕХНІЧНІ ВИМОГИ	3
5.1. Вимоги до розроблюваного продукту	3
5.2. Вимоги до програмного забезпечення	3
5.3. Вимоги до апаратного забезпечення	3
5.4. Вимоги до програмної документації	3
6. ЕТАПИ РОЗРОБКИ	4

					ДП 4671. 01.000 ТЗ		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробник		Якубовський А.Ю.			Система автоматизованого інформування іноземних студентів Технічне завдання	Літ.	Аркуш
Керівник		Павлов В.Г.					1
						Аркушів	4
Н/контр.		Сімоненко В.П.				НТУУ “КПІ ім. Ігоря Сікорського”, ФІОТ, ІО-62	
Зав. каф.		Стіренко С.Г.					

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дане технічне завдання розповсюджується на розробку автоматизованої системи інформування студентів.

Область застосування: практичне використання викладачами задля економії власного часу.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки служить завдання на виконання роботи кваліфікаційно-освітнього рівня «бакалавр комп'ютерної інженерії», затверджене кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний інститут ім. Ігоря Сікорського».

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою роботи є створення системи автоматизованого інформування іноземних студентів, оскільки саме ця група людей, частіше за все, обмежена у своїх можливостях вчасно отримувати важливу інформацію через мовний бар'єр, період адаптації в новій для них країні або звичайну невпевненість. Тому розробка даної системи одночасно вирішує дві проблеми: допомога іноземним студентам і оптимізація робочого процесу викладачів, шляхом автоматизації процесу інформування.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами для розробки служать науково-технічна література з інформаційних технологій, публікації в періодичних виданнях, а також відповідні статті в мережі Інтернет за даним питанням.

					ДП 4671. 01.000 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		2

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розроблюваного продукту

- Система повинна використовуватися лише в навчальних, освідомчо-інформативних цілях;
- Конкурентроspроможний рівень якості продукту;
- Зручність і простота використання розроблюваної системи.

5.2. Вимоги до програмного забезпечення

- Операційна система MS Windows XP, MS Windows 7, MS Windows 8/8.1, MS Windows 10, MacOS, Linux;
- Інтернет браузер Google Chrome, Mozilla Firefox, Opera.

5.3. Вимоги до апаратного забезпечення

- Комп'ютер на базі процесору Intel Pentium 4 / Athlon 64 і вище з підтримкою SSE2;
- Оперативної пам'яті не менше 512 Мбайт.

5.4. Вимоги до програмної документації

Програмною документацією до розроблюваної системи автоматизованого інформування іноземних студентів являється пояснювальна записка.

					ДП 4671. 01.000 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

6. ЕТАПИ РОЗРОБКИ

	Дата
Опрацювання науково-технічної літератури	11.03.2020
Формування вимог до розроблюваної системи	20.03.2020
Вибір засобів програмування	23.03.2020
Реалізація інформаційної системи	11.04.2020
Тестування та оптимізації отриманого продукту	30.04.2020
Оформлення документації дипломної роботи	01.05.2020
Робота над помилками, зауваженнями та порадами	25.05.2020

					ДП 4671. 01.000 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

Пояснювальна записка
до дипломного проєкту
на тему: «Система автоматизованого інформування
іноземних студентів»

Київ – 2020 року

ЗМІСТ

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ	3
ВСТУП.....	5
РОЗДІЛ 1 ОГЛЯД ІНФОРМАЦІЙНИХ СИСТЕМ І АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	7
1.1. Етапи розвитку інформаційних систем	7
1.2. Основні процеси, що протікають у інформаційних системах.....	9
1.3. Основні типи та класифікації інформаційних систем	10
1.4. Автоматизовані інформаційні системи	11
1.5. Аналіз існуючих рішень на основі програмного забезпечення «ePoshta Mailer».....	14
ВИСНОВКИ ДО РОЗДІЛУ 1	21
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА ПОСЛІДОВНІСТЬ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	22
2.1 Очікуваний функціонал і технічні характеристики системи	22
2.2 Головні сценарії взаємодії користувача з системою	24
2.2.1. Сценарій входу у систему	25
2.2.2. Сценарій внесення нових студентів	26
2.2.3. Сценарій створення нових скриптів	27
2.2.4. Сценарій налаштування профілю користувача.....	28
2.2.5. Сценарій відправлення електронних листів.....	29
2.3 Проєктування структури бази даних	30

					<i>ДП 4671. 02.000 ПЗ</i>		
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
Розробник		Якубовський А.Ю.			Система автоматизованого інформування іноземних студентів Пояснювальна записка	Літ.	Аркуш
Керівник		Павлов В.Г.					1
						Аркуші	62
Н/контр.		Сімоненко В.П.				НТУУ "КПІ ім. Ігоря Сікорського", ФІОТ, ІО-62	
Зав. каф.		Стіренко С.Г.					

2.4 Вибір технологій розробки	32
2.4.1 Вибір СУБД.....	32
2.4.2 Вибір технологій розробки серверної частини	32
2.4.3 Вибір технологій розробки користувацького інтерфейсу	34
2.5 Розробка програмного продукту	35
2.5.1 Розробка серверної частини	35
2.5.2 Розробка користувацького інтерфейсу	39
ВИСНОВКИ ДО РОЗДІЛУ 2	42
РОЗДІЛ 3 ТЕСТУВАННЯ ТА ОПИС ОТРИМАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ	43
3.1. Інструкція користувача	43
3.2. Моделювання роботи системи	57
ВИСНОВКИ ДО РОЗДІЛУ 3	59
ВИСНОВКИ	60
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	61

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ

ОС	Операційна система
ІС	Інформаційна система
АІС	Автоматизована інформаційна система
ООП	Об'єктно-орієнтоване програмування
БД	База даних
СУБД	Система управління базою даних
ПЗ	Програмне забезпечення
ЛКМ	Ліва кнопка миші
SQL	(англ. Structured query language) декларативна мова програмування для взаємодії користувача з базами даних БД База даних
DBF	(англ. Data Base File) формат збереження даних
SMTP	(англ. Simple Mail Transfer Protocol) мережевий протокол, призначений для передачі електронної пошти
SSL	(англ. Secure Sockets Layer) протокол захисту транспортного рівня
TLS	(англ. Transport Layer Security) протокол захисту транспортного рівня
HTML	(англ. HyperText Markup Language) мова тегів, яку використовують для створення гіпертекстових документів у мережі Інтернет
UML	(англ. Unified Modeling Language) уніфікована мова моделювання
ORM	(англ. Object-Relational Mapping) технологія програмування, яка пов'язує бази даних з концепціями об'єктно-орієнтованих мов програмування

CRUD	(англ. Create, Read, Update, Delete) акронім, який позначає базові функції при роботі з базою даних
URL	(англ. Uniform Resource Locator) система уніфікованих адрес електронних ресурсів
MVC	(англ. Model-View-Controller) схема розподілення даних додатку, користувацького інтерфейсу та керуючої логіки на три окремих компоненти: модель, уявлення, контролер
IoC	(англ. Inversion of Control) принцип подубови програми, при якому її частини отримують потік керування із загальної бібліотеки
SPA	(англ. Single-Page Application) односторінковий застосунок
CSS	(англ. Cascading Style Sheets) спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду

ВСТУП

Сучасна світова тенденція показує, що більшість наявних технологій прагнуть автоматизувати ту чи іншу частину роботи, яку ми виконуємо щодня, навіть не замислюючись над цим. Ми також вирішили не проходити повз цього тренду та створити інформаційну систему, яка за допомогою наперед створених скриптів, інформувала б іноземних студентів шляхом автоматизованої розсилки листів на електронну пошту.

Актуальність теми

У теперішній час гостро стоїть проблема раціонального використання власного часу, або як це ще називають «тайм менеджменту». Тому наш програмний застосунок призначений задля економії часу викладачів, які замість рутинної роботи могли б потратити його на підвищення власної кваліфікації або ж створення цікавого та інтерактивного навчального матеріалу.

Мета і задачі дослідження

Метою роботи є створення системи автоматизованого інформування іноземних студентів, оскільки саме ця група людей, частіше за все, обмежена у своїх можливостях вчасно отримувати важливу інформацію через мовний бар'єр, період адаптації в новій для них країні або звичайну невпевненість. Тому розробка даної системи одночасно вирішує дві проблеми: допомога іноземним студентам і оптимізація робочого процесу викладачів, шляхом автоматизації процесу інформування. Але варто відмітити, що розроблена система не обмежується інформуванням лише іноземних студентів, а є універсальним інструментом для розповсюдження важливої та необхідної інформації для будь-якої категорії студентів.

Для досягнення поставленої мети були виокремлені наступні основні задачі:

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

- Провести порівняльний аналіз уже готових, схожих за ідеєю програмних рішень;
- Створити ефективну базу даних, яка б максимально швидко б обробляла вхідні запити;
- Створити готовий програмний продукт на основі запропонованих вимог;
- Проаналізувати отримані результати та зробити відповідні висновки щодо всього проєкту.

Практичне значення

На нашу думку, дана інформаційна система повинна користуватися великим попитом серед викладачів, які не бажають кожного разу витрачати купу свого дорогоцінного часу на виконання рутинної роботи, яку після виконання первинної підготовки та налаштування, можна зробити в один клік. Тому ми вважаємо, що створена нами система містить високе практичне значення та потенціал.

					ДП 4671. 02.000 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1

ОГЛЯД ІНФОРМАЦІЙНИХ СИСТЕМ І АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Система – це сукупність певних елементів, які при взаємодії один з одним створюють одне ціле.

Інформаційна система – це взаємозв'язок між комплексом засобів, методів і людських ресурсів, який використовується задля збереження, обробки та видачі інформації з метою досягнення поставлених цілей[1].

Сучасне представлення інформаційної системи кожного з нас передбачає, що головну роль для функціонування цілої системи повинен відігравати основний технічний засіб для обробки інформації – комп'ютер. Але потрібно розуміти, що комп'ютер – це, насамперед, інструмент, який містить спеціалізовані програмні додатки, які без відповідного персоналу і засобів зв'язку та комунікації, не зможуть функціонувати, а тим більше – являти собою цілу ІС.

1.1. Етапи розвитку інформаційних систем

Перші ІС з'явилися на початку 50-х років і були призначені, вцілому, для ведення бухгалтерського обліку, по типу розрахунку заробітної плати чи обробки різноманітного роду рахунків. Ці всі задачі реалізовувалися на електромеханічних лічильних машинах, що значно економило час і затрати на підготовку бумажних документів.

У шестидесяті роки повністю змінився підхід до використання ІС. Люди почали застосовувати інформацію, отриману від ІС, для досягнення практичної вигоди у різноманітних сферах життя. Це сприяло розвитку комп'ютерного обладнання для широкого застосування, яке було б спроможне виконувати низку нових функцій, а не лише ведення бухгалтерського рахунку.

Семидесяті та вісімдесяті роки можна охарактеризувати, як період, коли все більше і більше великих організацій впроваджували інформаційні системи у робочий процес, задля того, щоб прискорити процес прийняття рішень.

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

У кінці вісімдесятих – початку дев'яностих років, концепція використання ІС знову змінюється та стає тою, яку ми знаємо і сьогодні. Вони стають головним стратегічним джерелом інформаційних ресурсів і починають використовуватися практично на всіх рівнях організації у будь-якій сфері роботи. ІС цього періоду починають автоматизовувати, задля того щоб мінімізувати людські ресурси, необхідні задля коректної роботи всієї системи[2].

У таблиці 1.1 зображені головні етапи розвитку інформаційних систем і цілі, які вони дотримуються.

Таблиця 1.1.

Головні етапи розвитку ІС

Період часу	Концепція використання інформації	Вид інформаційних систем	Ціль використання
1950-1960 роки	Потік розрахункових документів, які знаходяться в бумажному виді	Електромеханічні бухгалтерські машини	Спрощення процедури обробки рахунків і розрахунку заробітної плати
1960-1970 роки	Допомога у підготовці звітів	Різноманітні інформаційні системи управління	Прискорення процесу підготовки звітності
1970-1980 роки	Контроль різноманітних процесів з метою їх подальшого управління	Системи підтримки прийняття рішень	Вироблення практичних, раціональних і ефективних рішень
1980 рік – наш час	Представлення інформації у вигляді стратегічного ресурсу	Стратегічні ІС. Зародження автоматизованих систем	Задоволення інформаційних потреб споживачів і покращення показників ефективності

1.2. Основні процеси, що протікають у інформаційних системах

Інформаційний процес – це проведення певних дій над інформаційними ресурсами (даними), а саме: створення, збирання, зберігання, обробка, відображення, передавання, розповсюдження, використання, захист, знищення та інші.

В загальному, процеси, які забезпечують роботу будь-якої ІС можна умовно розділити на наступні частини[3]:

- Ввід інформації, який може бути реалізований через внутрішні або ж зовнішні джерела;
- Обробка вхідної інформації та представлення її в зручному для системи виді (процес конвертування даних);
- Вивід інформації кінцевому споживачу, який може бути як і клієнтом системи – людиною, яка використовує ІС задля задоволення власних певних потреб, так і іншою системою, яка використовує дані цієї ІС задля досягнення власних цілей;
- Feedback, або ж зворотній зв'язок – інформація, яка надається переважно користувачами ІС, задля покращення її коректної роботи.

На рис. 1.1. зображено основні процеси, які протікають у інформаційних системах і взаємозв'язки між ними.

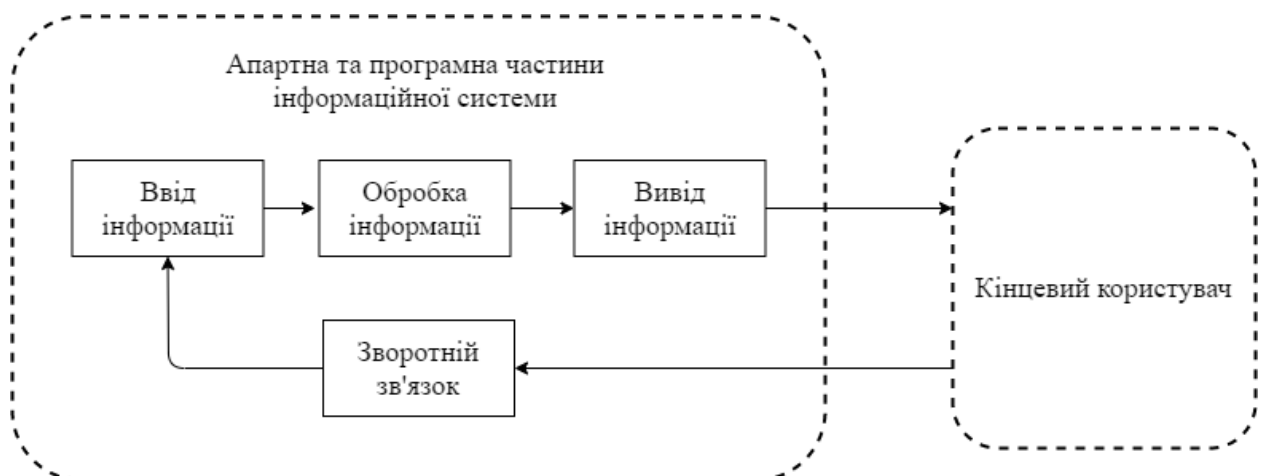


Рис. 1.1. Основні процеси в ІС

1.3. Основні типи та класифікації інформаційних систем

Тип інформаційної системи, в першу чергу, залежить від того, які цілі вона переслідує та чийм інтересам вона підпорядковується. По характеру представлення і організації збережених даних ІС поділяють на фактографічні, документальні та геоінформаційні[4].

Фактографічні ІС накопичують і зберігають інформаційні одиниці у вигляді множини екземплярів одного чи декількох типів структурних елементів (інформаційних об'єктів). Кожен з цих елементів відображає відомості на рахунок певного факту чи події, незалежно від всіх інших представлених фактів.

Однією з особливостей фактографічних ІС є те, що наповнення інформаційної бази в цих системах передбачає, зазвичай, обов'язковий процес структуризації даних. Дані інформаційні системи забезпечують задоволення потреб користувачів шляхом надання останнім актуальних відомостей, таких як факти, конкретні показники з тої чи іншої сфери діяльності, тощо.

В документальних ІС одиницею інформації виступає нерозподілений на менші частини документ, а вхідна інформація, як правило, або повністю не підлягає структуризації, або структурується лише частково.

Деякі види документальних інформаційних систем забезпечують встановлення логічного зв'язку вхідних документів. Це дозволяє посортувати велику кількість інформації, яка представлена у вигляді документів, по змісту роботи, кількості сторінок, використаним джерелам, тощо та надати її у зручному виді кінцевим споживачам.

В геоінформаційних ІС дані представлені у виді окремих інформаційних об'єктів, які відносяться до загальної електронної топографічної основи, якою, зазвичай, виступає електронна карта. Даний тип систем використовується у тих випадках, коли структура інформаційних об'єктів і процесів містить просторово-географічний компонент.

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

Класифікація інформаційних систем по способу організації архітектури[5]:

- ІС на основі архітектури файл-сервер;
- ІС на основі архітектури клієнт-сервер;
- ІС на основі багаторівневої архітектури;
- ІС на основі інтернет або інтернат технологій.

Також, одним з ключових факторів, за яким характеризують інформаційні системи є їхній степінь автоматизації:

- Ручні ІС, для яких характерна відсутність сучасних технологічних засобів обробки інформації та виконання усіх потрібних операцій, безпосередньо, людиною;
- Автоматичні ІС – системи, які виконують повністю усі операції по переробці та аналізу інформації, без будь-якого втручання зі сторони людини.
- Автоматизовані ІС припускають участь в обробці інформації, як і людини, так і технічних обчислювальних машин, при цьому головна роль відводиться, звичайно, комп'ютеру.

Саме під поняттям автоматизованої ІС, люди розуміють термін «інформаційна система» вцілому, ототожнюючи та співставляючи ці два визначення. Це спричинене тим, що метод обробки інформації саме автоматизованих ІС використовується в більшості систем, які мають багатий практичний функціонал і широку сферу застосування. Тому, саме системи автоматизованого характеру, ми, як кінцеві споживачі, використовуємо найчастіше.

1.4. Автоматизовані інформаційні системи

Основною метою автоматизації будь-якої інформаційної системи є підвищення ефективності та продуктивності роботи працівників, покращення якості продуктів і послуг, які ця система надає, а також збільшення

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

максимальної кількості кінцевих користувачів, яких ця ІС може одночасно обслуговувати, або, як це ще називають – процес масштабування. Автоматизація інформаційних процесів, які відбуваються в ІС здійснюється за допомогою засобів обчислювальної техніки та відповідного програмного забезпечення.

До основних принципів автоматизації інформаційних процесів відносять: окупність, надійність, безпека, гнучкість і відповідність загальним стандартам якості.

Основні задачі автоматизації ІС полягають у:

- Зменшенні кількості рутинної роботи;
- Зменшенні часових затрат на виконання традиційних інформаційних процесів;
- Прискорення швидкості обробки великої кількості інформаційних ресурсів;
- Розширення можливостей здійснення статистичного аналізу;
- Збільшення точності при виконанні складних обчислень і при веденні обліково-звітної інформації;
- Збільшення максимальної кількості кінцевих споживачів;
- Підвищення оперативності та рівня якості обслуговування кінцевих користувачів;
- Модернізація або повна заміна елементів традиційних технологій;
- Розширення можливостей ефективного використання та організації інформаційних ресурсів, шляхом впровадження сучасних і актуальних технологій;
- Поширення великих обсягів вихідної інформації між користувачами ІС або ж іншими автоматизованими системами, які тісно взаємодіють між собою;
- Оптимізація всього робочого процесу, починаючи від вводу даних і закінчуючи видачою отриманої інформації.

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

Отже, автоматизована інформаційна система – це система, яка складається з відповідних технічних засобів, програмного забезпечення та персоналу, які утворюють між собою комплекс засобів автоматизації інформаційних процесів з метою реалізації встановлених функцій і досягнення поставлених цілей[6].

Виділяють наступні типи АІС по масштабу застосування:

- Охоплюють один процес у одній організації;
- Охоплюють певну множину процесів у одній організації;
- Забезпечують функціонування одного процесу в масштабі декількох взаємопов'язаних організацій;
- Реалізують роботу великої кількості інформаційних процесів у масштабі низки організацій, які пов'язані між собою.

Основною метою АІС являється забезпечення ефективного алгоритму збереження, накопичення, обробки та видачі великої кількості інформаційних ресурсів по відповідним запитам зі сторони кінцевих користувачів, спричиненого задля найбільш повного задоволення інформаційних потреб останніх.

Також, варто розуміти, що задля коректного функціонування АІС, процедури збору та вводу даних є обов'язковими, адже, зазвичай, вся необхідна та достатня для роботи системи вхідна інформація може знаходитися у складі її бази даних. Наявність цих процедур у АІС залежить, скоріше за все, від тонкощів і спеціалізації самої системи, її функціоналу та сфери, в якій вона задіяна.

АІС можуть бути достатньо простими (елементарні інформаційні довідники) і складними системами (системи прогнозування, які залежать від великої кількості факторів). Але навіть прості АІС мають багатозначні структурні відношення між своїми елементами, модулями та іншими складовими. Ця обставина дозволяє віднести їх до класу складних систем, що складаються з взаємопов'язаних частин, які працюють у складі однієї цілісної багаторівневої структури[7].

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

На рис 1.2. зображено основні компоненти та технологічні процеси, які відбуваються у автоматизованих інформаційних системах.

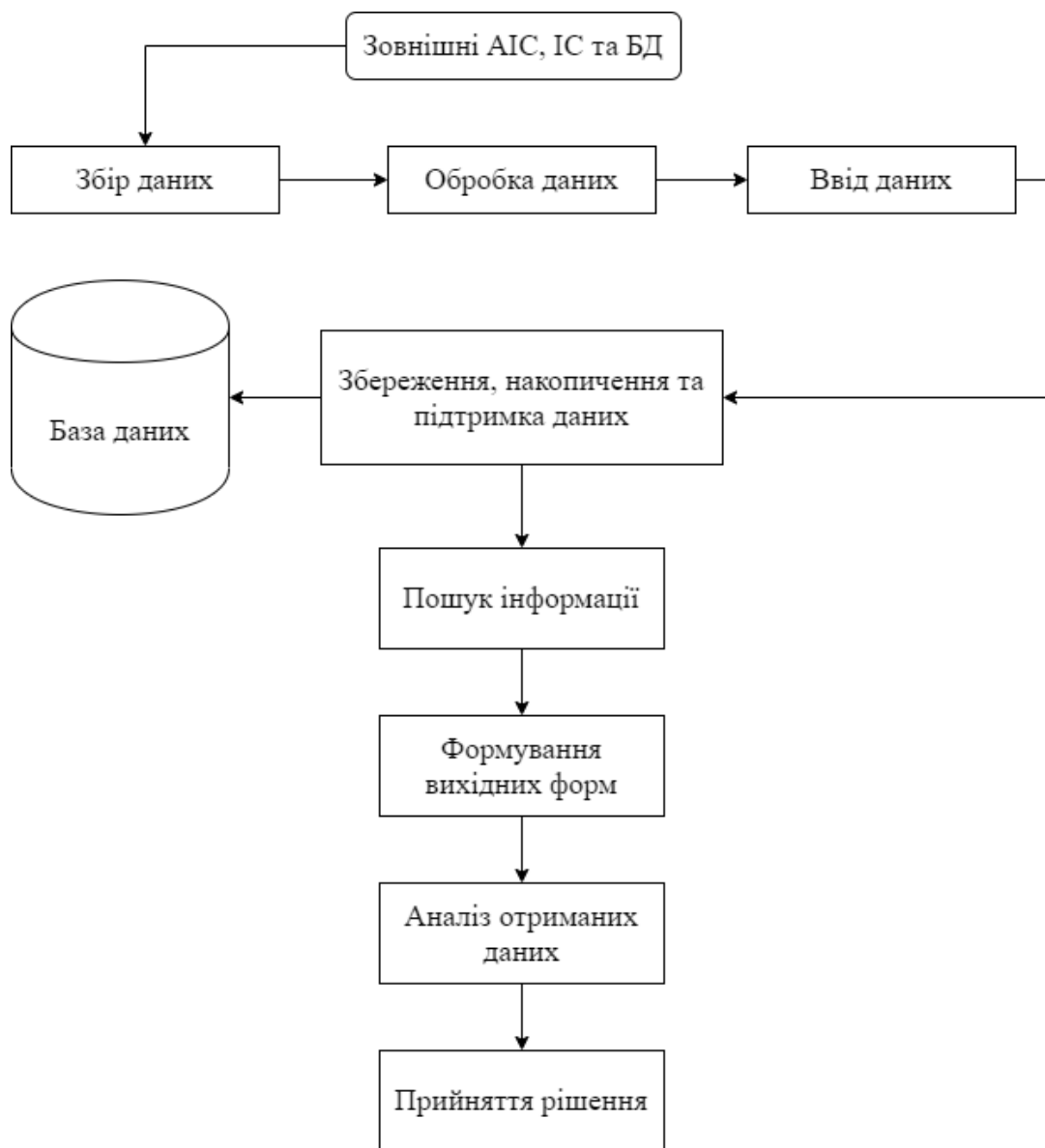


Рис 1.2. Основні компоненти та процеси АІС

1.5. Аналіз існуючих рішень на основі програмного забезпечення «ePoshta Mailer»

Сьогодні, електронна пошта широко застосовується багатьма людьми з метою швидкого поширення великої кількості інформації. Тому сервіси, які надають можливість відправлення масових електронних листів, користуються великою популярністю серед маркетологів великих компаній.

Провівши огляд існуючих рішень, було виявлено чималу кількість схожих сервісів по відправці електронних листів, які конкурують один з одним, а саме: «eSputnik», «ePoshta Mailer», «Unisender». Дані сервіси мало в чому відрізняються один від одного, в основному, це відмінність у технічних характеристиках і невелика різниця у функціоналі. Тому розглянемо найбільш комплексне програмне забезпечення – «ePoshta Mailer», яке містить і охоплює найбільшу кількість можливостей серед усіх представлених варіантів.

Як зазначають самі розробники програми, «ePoshta Mailer» – професійна та ефективна програма для відправлення масових email кампаній. Основні можливості цього інструменту становлять створення і відправка великої кількості email повідомлень на список розсилки будь-якого обсягу.

Розглянемо основні можливості, які надає ця програма.

Необмежений список email адрес

Список розсилки може включати велику кількість email адрес і полів персоналізації, які необхідні для використання. Також присутня можливість імпорту списку адрес за наступними способами:

- Із файлу (*.csv, *.txt, *.xls, *.xlsx, *.doc, *.docx, *.mdb, *.accdb);
- Із бази даних DBF;
- Із буфера обміну;
- Із адресної книги Outlook;
- Із іншим програм лінійки «ePoshta» для збору email адрес з Інтернету, локальних файлів та бази даних WHOIS;

Підтримка SMTP серверів

Програма підтримує роботу з трьома типами SMTP серверів:

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

- Вбудований сервер, який має максимальну швидкість відправлення листів і застосовується у тому випадку, коли провайдер не забороняє пряме надсилання;
- Зовнішній SMTP сервер, який і використовується тоді, коли провайдер не дозволяє пряму адресацію електронних листів. Головною проблемою роботи з цим типом серверів є те, що користувач повинен самостійно знайти зовнішній SMTP сервер для розсилки повідомлень;
- «ePoshta» SMTP сервер, який можна швидко інтегрувати у будь-яке програмне забезпечення для відправлення великої кількості електронних листів з мінімальними налаштуваннями та затратами по часу.

Підтримка Proxu серверів

Програма дозволяє підтримувати роботу з низкою проху серверів. Перед початком роботи, кожен сервер проходить перевірку, що допомагає виявити та видалити недієспроможні сервери зі списку.

Підтримка SSL/TLS аутентифікації

Програмне забезпечення «ePoshta Mailer» відправляє електронні листи, використовуючи безпечне Інтернет-з'єднання. Застосування SSL/TLS протоколів аутентифікації у програмі, гарантує безпечне та захищене відправлення електронних листів. Завдяки використанню даної технології можливе повторне підключення без повторної аутентифікації. Основною функцією SSL/TLS протоколів є зашифрувати або ж сховати важливу конфіденційну інформацію (паролі, стратегічні кампанії та показники, тощо) з метою її збереження та не потрапляння в чужі руки у процесі відправлення повідомлень.

Підтримка віджетів з соціальних мереж

Дана можливість використовується задля того, щоб кінцевий користувач системи міг без проблем додати зацікавивший його віджет популярних соціальних мереж, наприклад Facebook або Twitter, до змісту відправленого

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

повідомлення. Зазвичай, ця функція використовується задля того, щоб залучати нових людей до потрібних вам профілей соціальних мереж. Це реалізується за допомогою мініатюрних іконок, які містять посилання на необхідний профіль. Варто також відмітити те, що зміст електронних листів, які містять дані віджети, за статистикою рідше попадають у папку «спам» цільових електронних скриньок під час відправлення величезної кількості повідомлень.

Підтримка методу «Посилання відмови»

Даний метод необхідний для того, щоб одержувач email повідомлення міг у будь-який момент відмовитися від даного типу розсилки, при цьому зберігаючи лояльність клієнтів. Програмне забезпечення «ePoshta Mailer» включає два стандартних типи відмови:

- Відмова по email. В цьому випадку користувач повинен самостійно вказати email адреси, на які надійдуть повідомлення про відмову від розсилки;
- Відмова через веб-форму на веб-додатку. Цей спосіб зручний тим, що у текст повідомлення додається гіперпосилання на сторінку з формою відмови. Користувачу не прийдеться власноруч оновлювати поточний список розсилки, адже вбудований алгоритм буде автоматично виконувати цю функцію, видаляючи адреси тих одержувачів, хто відмовився від розсилки.

Персоналізація змісту листів

Дана функція надає можливість динамічно змінювати певні показники, які заздалегідь записані у базі даних. Персоналізація дозволяє підставляти потрібне готове значення на вказане місце у структурі електронного листа, підставляючи його у загальну форму (наприклад, поле {Name} буде замінено на ім'я та прізвище отримувача). Ця функція значно спрощує масову розсилку електронних повідомлень, автоматизуючи весь процес надсилання листів. Персоналізація email повідомлень, також, значно підвищує їх шанс на

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

проходження через спам-фільтри різноманітних сервісів, що надають можливість відправляти електронні листи.

Спін-текст

Дана функція необхідна, щоб успішно проходити перевірку на спам шляхом заміною ряду слів на синонімічні. Користувач має змогу самостійно додати синоніми до потрібних йому слів, які в кінцевому результаті будуть випадково використовуватися під час відправлення електронних листів, тим самим збільшуючи відсоток email повідомлення, які успішно пройдуть найпоширеніші спам-фільтри.

Спам-тест

Програмне забезпечення «ePoshta Mailer» містить вбудований спам-фільтр, який перед відправленням email повідомлень оцінює та аналізує його вміст і виставляє певні бали відповідно до рівня спаму. При високій оцінці спаму, вбудований алгоритм самостійно попередить користувача та вкаже на помилки, які йому слід виправити.

HTML і текстовий редактори

Дана програма підтримує роботу як і з текстовим, так і з HTML форматом повідомлення. Головним інструментом для написання та відправлення листів, на сьогоднішній момент, являється HTML редактор, адже він є дуже популярним і підтримується багатьма популярними сервісами електронної пошти. Текстовий редактор потрібен лише в тому випадку, коли старі поштові клієнти не мають фізичної змоги підтримувати HTML формат. В загальному, текстовий редактор має набагато скудніший функціонал, але його також можна використовувати при написанні листів, адже вся текстова частина автоматично синхронізується з HTML частиною. При цьому, саме HTML редактор містить повноцінний функціонал, включаючи стандартні опції з вибору розміру та типу шрифтів, вставлення таблиці, зображення чи

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

гіперпосилання на необхідні джерела, тощо. Також, варто відмітити можливість прикріплення файлів для будь-якого з режимів роботи.

Моніторинг розсилки

Дана можливість дозволяє слідкувати за статистикою відправлених електронних листів, а саме: кількість відкритих і прочитаних повідомлень, час прочитання одного листа, кількість переходів за надісланими посиланнями, кількість повідомлень, які не надійшли до кінцевого отримувача за тієї чи іншої причини, тощо. Також, програма підтримує роботу з популярним на сьогодні сервісом по накопиченню різноманітних статистичних показників – Google Analytics.

Детальні звіти

Після відправлення масової розсилки, програма «ePoshta Mailer» може генерувати два типи звітів:

- Індивідуальний;
- Груповий з урахуванням додаткових критеріїв.

Усі звіти можна експортувати та зберігати локально на ком'ютері чи іншому пристрої збереження даних.

Керування списками розсилки

Даний програмний комплекс оснащений вбудованими засобами управління списками електронних адрес, а саме:

- Додавання та видалення контактів;
- Сортування списку по заданому критерію
- Редагування інформації у списку;
- Видалення підозрілих і ідентичних адрес.

Отже, проаналізуємо весь представлений функціонал програмного забезпечення «ePoshta Mailer» і визначимо конкретні переваги та недоліки цієї автоматизованої інформаційної системи.

Таблиця 1.2.

Переваги та недоліки ПЗ «ePoshta Mailer»

Переваги	Недоліки
Багатий функціонал для редагування змісту повідомлень.	Ціна, адже дане програмне забезпечення не являє собою безкоштовний продукт.
Розширені можливості по успішному подоланню спам-фільтрів різноманітних поштових клієнтів.	Громіздкий, складний і застарілий користувацький інтерфейс, що створює чималий поріг входу для нових користувачів.
Великий вибір готових серверів, через які можна здійснювати відправлення великої кількості повідомлень з дуже високою швидкістю.	Відсутня можливість створювати власні, гнучко настроювані сценарії та шаблони листів, які автоматично будуть відправлятися за кожний N період часу.
Баунси – зворотнє повідомлення з інформацією про некоректну або недійсну email адресу, яка відправляється сервером, безпосередньо, адресату.	Потребує значних зусиль для первинного налаштування усіх автоматизованих процесів і функціонування всієї системи в загальному.

ВИСНОВКИ ДО РОЗДІЛУ 1

1. Автоматизована інформаційна система – це система, яка складається з персоналу та комплексу засобів автоматизації її діяльності, які реалізують автоматизовану технологію виконання передбачуваних заздалегідь функцій.

2. Порівняльний аналіз існуючих рішень виявив, що вони не дають змогу повністю вирішити поставлені завдання та проблеми, які існують наразі. Тому було прийнято рішення створити власну автоматизовану систему інформування іноземних студентів, яка буде чітко, а головне – швидко виконувати задачі, які на неї покладають.

					ДП 4671. 02.000 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА ПОСЛІДОВНІСТЬ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

З метою створення конкурентноспроможної інформаційної системи потрібно заздалегідь передбачити увесь прогнозований функціонал, чітко визначити цільову аудиторію даної програми та проблеми, які вона повинна вирішувати, а також вибрати відповідні засоби для розробки ПЗ, які будуть якомога краще виконувати поставлені задачі та відповідати сучасним стандартам якості. Даний комплекс дій допоможе зосередитися на по-етапній розробці системи та сконцентрувати усі свої вміння та навички на реалізації поставлених функцій.

2.1 Очікуваний функціонал і технічні характеристики системи

Прогнозуються наступні технічні характеристики системи:

- Автоматизована інформаційна система буде реалізована у формі веб-додатку, задля того, щоб будь-який користувач мав можливість отримати доступ до програми з будь-якого пристрою, який підтримує основні найпопулярніші веб-браузери – Mozilla, Opera, Chrome;
- Автоматизована система інформування іноземних студентів буде оснащена зрозумілим, «дружнім» до користувача інтерфейсом, який буде мати низький поріг входу та інтуїтивну структуру розположення ключових елементів управління;
- Програмний комплекс буде мати просту структуру організації, де в будь-який момент можна буде додати новий функціональний елемент;
- АІС буде взаємодіяти з актуальною на сьогодні системою управління бази даних – MySQL, яка широко застосовується на ринку та є однією з найкращих серед існуючих рішень для керування БД, які містять великі об'єми інформації;

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

Передбачається наступний перелік доступних функцій:

- Реалізація операції входу на власний аккаунт (процес авторизації на веб-сайт) і операції створення нового аккаунту, якщо такий відсутній;
- Можливість редагувати списки студентів, додавати нові значення та показники, безпосередньо, на головній сторінці веб-додатку;
- Можливість імпортувати списки студентів з відповідними даними з excel файлів;
- Можливість створення власного шаблону, який буде автоматично використовуватися при створенні нового електронного листа, тому не прийдеться кожного разу писати інформацію, яка повторюється з разу в раз;
- Можливість створення власних, гнучких і легко-настроюваних сценаріїв автоматичного відправлення email повідомлень, для яких можна буде здійснити вибір періоду надсилання й обрати необхідні критерії, відносно яких будуть формуватися списки студентів для розсилки та формуватися зміст, безпосередньо самого, електронного листа;
- Реалізація технології баунсу, коли відповідний сервер надсилає повідомлення на пошту адресату у разі, коли поштова скринька отримувача недоступна, або взагалі не існує;
- Можливість сортувати відповідні стовбці таблиці, задля моніторингу актуальної інформації;
- Реалізація текстового редактору з використанням персоналізованих і унікальних даних. Даний алгоритм дозволяє автоматично замінювати сталий показник на відповідне значення з таблиці користувача (наприклад, значення {groupName} та {studentId} будуть автоматично замінені на відповідно – ім'я,

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

прізвище та студентський квиток кожного студента, якому буде відправлено електронний лист).

2.2 Головні сценарії взаємодії користувача з системою

Для опису усіх головних процесів, які відбуваються в системі, будемо використовувати UML-діаграми діяльності, або як їх ще називають, діаграми активності. Даний тип діаграм використовується не лише для візуалізації динамічного характеру системи, але й для побудови різноманітного роду інформаційних систем за допомогою методів прямого та зворотнього проєктування[8].

Діаграма діяльності зображує потік управління від початкової до кінцевої точки, показуючи усі можливі шляхи прийняття рішень, які виникають у користувача під час використання певної системи. Ми можемо зобразити, як послідовну, так і паралельну обробку діяльності, використовуючи даний тип UML-діаграм.

Отже, UML-діаграми активності використовуються задля того, щоб:

- Динамічно зобразити роботу системи або конкретного процесу;
- Моделювати програмні елементи, такі як методи, операції та функції;
- Показати обмеження, умови та логіку алгоритмів;

Розглянемо декілька основних сценаріїв для користувача, які можуть виникнути в ході експлуатації даної автоматизованої інформаційної системи, а саме:

- Вхід у систему;
- Внесення нових студентів;
- Створення нових скриптів;
- Налаштування профілю користувача;
- Відправлення електронних листів певній групі студентів.

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

2.2.1. Сценарій входу у систему

Першою взаємодією будь-якого користувача з системою є, безпосередньо, вхід до цієї системи, який, зазвичай, реалізований за допомогою введення електронної пошти та паролю.

На рис 2.1. зображена UML-діаграма, яка описує всі можливі випадки при вході користувача у систему.

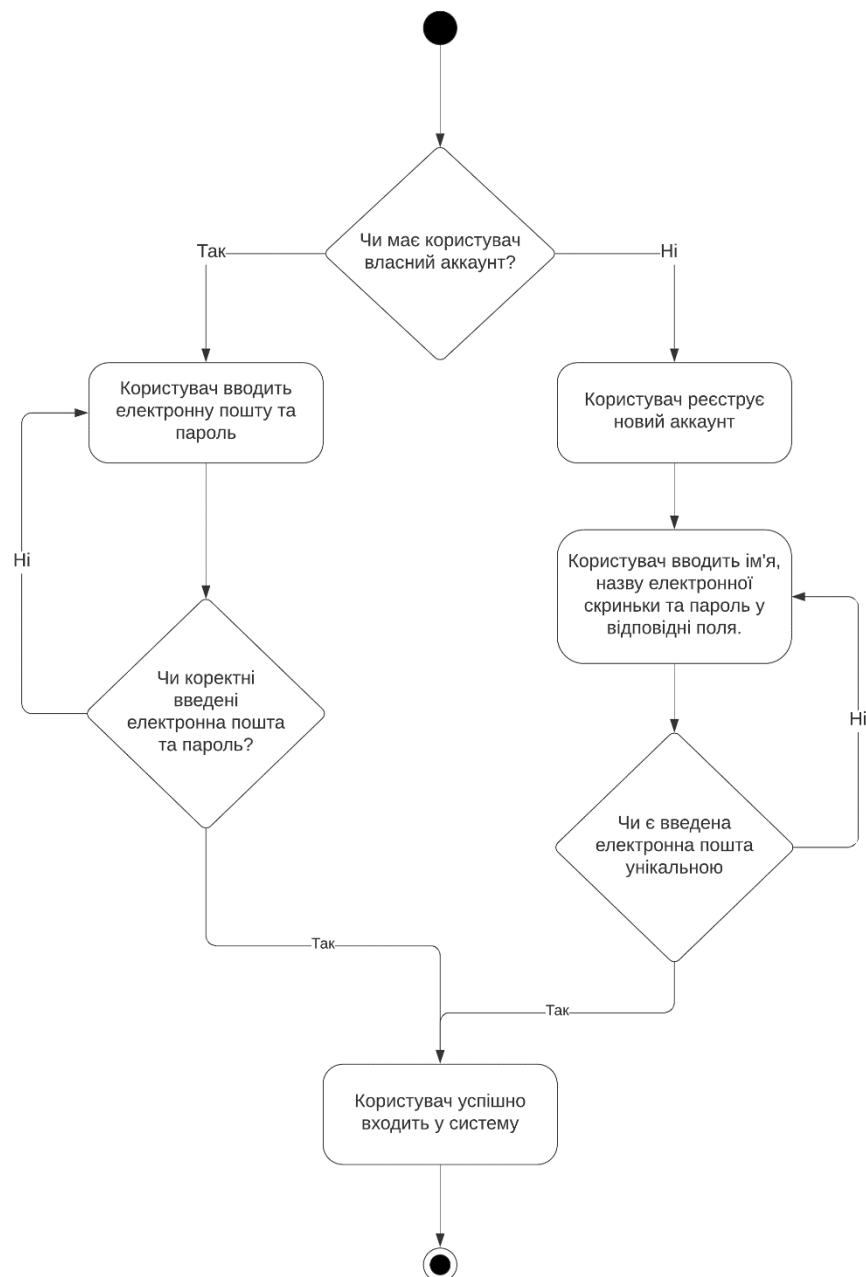


Рис. 2.1. UML-діаграма входу в систему

2.2.2. Сценарій внесення нових студентів

На головній сторінці, на яку користувач попадає відразу після проходження авторизації, він натискається на таблицю, в яку можна додати студентів, унікальні номери їхніх студентських квитків, номер групи, тощо.

На рис 2.2. зображена UML-діаграма, яка описує всі можливі випадки при внесенні нових студентів у систему.

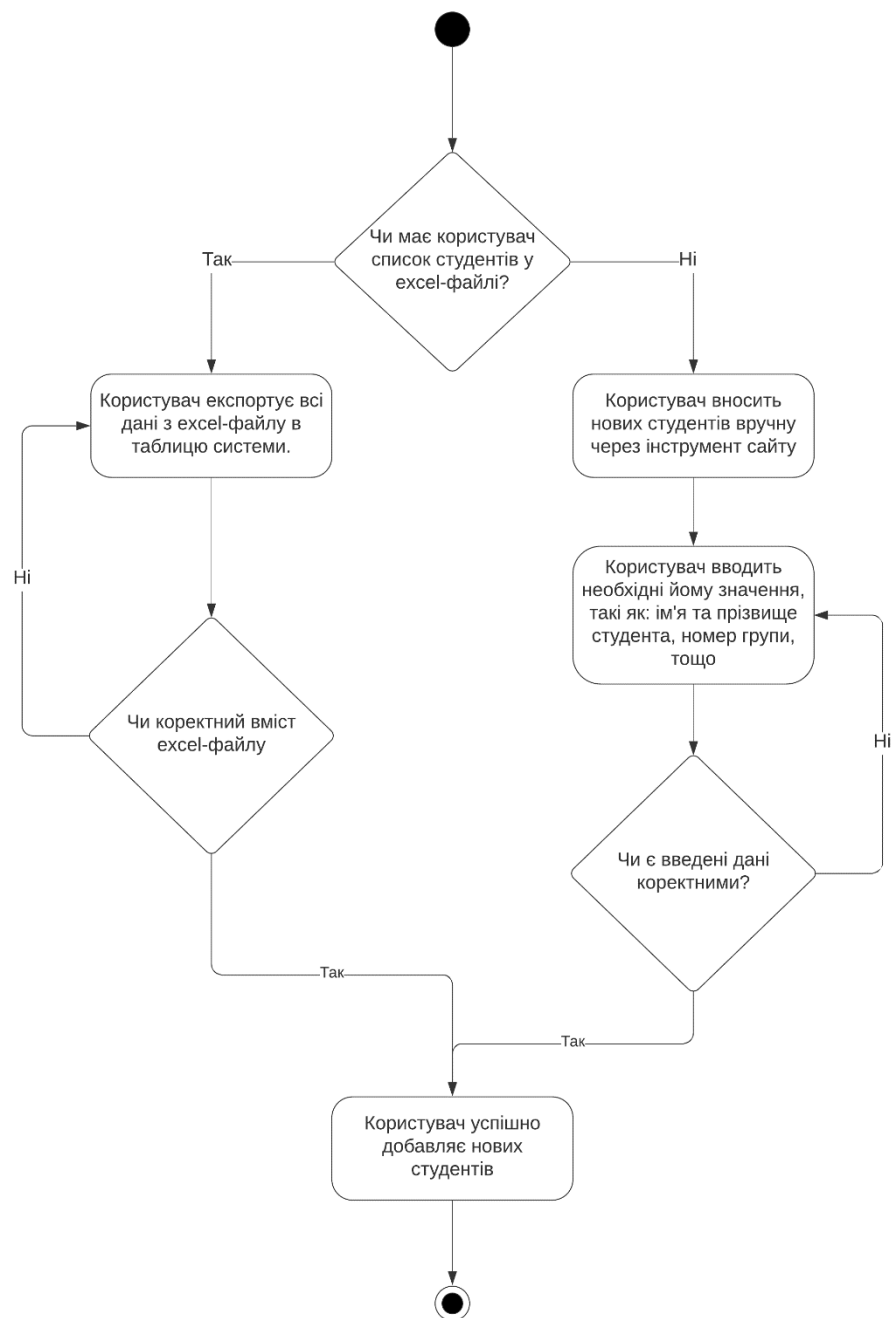


Рис 2.2. UML-діаграма внесення нових студентів

2.2.3. Сценарій створення нових скриптів

Також, користувач має змогу створити власні гнучко-настроювані скрипти, які значно автоматизують весь процес відправки великої кількості електронних листів.

На рис 2.3. зображена UML-діаграма, яка описує процес створення нових скриптів.

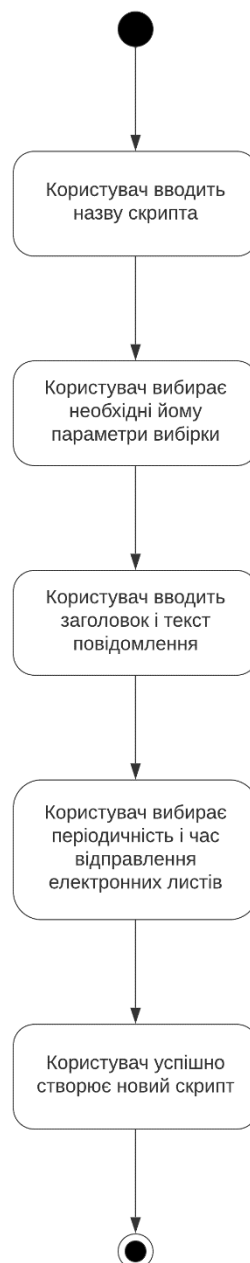


Рис 2.3. UML-діаграма створення нового скрипта

2.2.4. Сценарій налаштування профілю користувача

Під час процесу експлуатації системи, користувач може без проблем зайти у відповідне меню налаштувань і змінити необхідні йому параметри, задля підвищення зручності використання.

На рис 2.4. зображена UML-діаграма, яка описує процес налаштування профілю користувача.



Рис 2.4. UML-діаграма налаштування профілю користувача

2.2.5. Сценарій відправлення електронних листів

Основним завданням проєктованої інформаційної системи є відправлення великої кількості електронних листів і максимальна автоматизація цього процесу.

На рис 2.5. зображена UML-діаграма, яка описує процес відправлення електронних повідомлень.

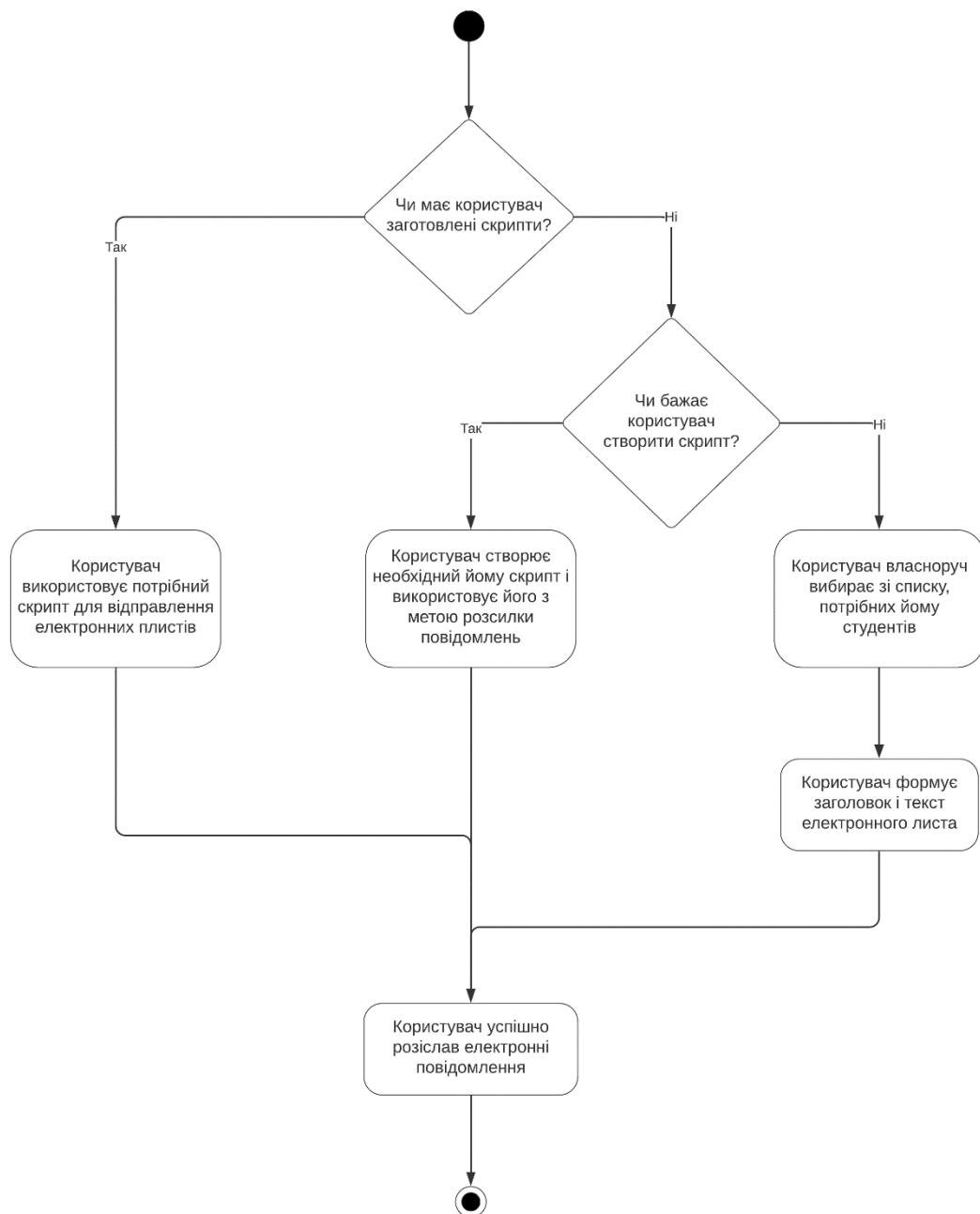


Рис 2.5. UML-діаграма відправлення електронних листів

2.3 Проектування структури бази даних

Основними об'єктами системи є користувач, інформація про студента та скрипти.

Для користувача було визначено наступні властивості:

- ідентифікатор, для визначення унікальності об'єкту та побудову зв'язків між ним та залежними об'єктами;
- електронна пошта, для підтвердження створення облікового запису користувача, та зворотнього зв'язку з ним у разі потреби;
- пароль, використовується для захисту облікового запису користувача. Для збереження в базі було використано алгоритм шифрування VCrypt;
- підпис, автоматично додається вкінці листа. Наприклад, може містити інформацію про особистість користувача та його посаду.

Інформація про студента являється залежним від користувача об'єктом та має зв'язок із ним багато-до-одного[9]. Було визначено наступні властивості:

- ідентифікатор, для визначення унікальності об'єкту;
- власник, користувач який вніс інформацію про студента;
- ім'я;
- електронна пошта;
- номер студентського;
- номер групи;
- заборгованість.

Скрипти дозволяють автоматизувати сценарій пошуку потрібних студентів та персоналізації повідомлень. Скрипти мають наступні властивості:

- ідентифікатор, для визначення унікальності об'єкту;

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

- власник, користувач який створив скрипт (зв'язок багато-до-одного);
- назва, для короткого опису скрипта;
- заголовок повідомлення, яке буде відправлено студентам, що відповідають заданим критеріям вибору;
- текст повідомлення, можлива персоналізація тексту за рахунок підстановки властивостей студента замість відповідних тегів у тексті (наприклад, {name});
- критерії, містять такі властивості, як назва поля за яким здійснюється фільтрація, умова та значення. Виділені в окрему таблицю зі зв'язком багато-до-одного у відношенні до скрипта.

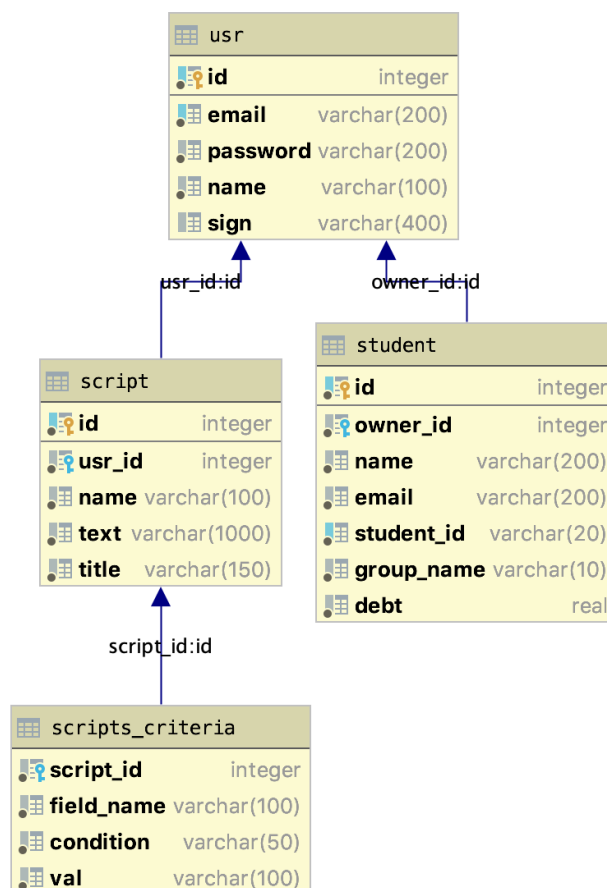


Рис. 2.6. Діаграма структури бази даних

2.4 Вибір технологій розробки

2.4.1 Вибір СУБД

В якості СУБД було обрано MySQL. Ключовими факторами вибору стали простота встановлення та налаштування, а також безпечність та швидкість роботи. Програмний продукт розповсюджується за ліцензією *GNU General Public License v3 (GPL-3)*, яка дозволяє вільне використання при вказанні копірайту. Також було розглянуто такі СУБД, як SQLite и PostgreSQL.

SQLite це файлова СУБД, вся база даних якої знаходиться в одному файлі, що робить зручним її використання, як вбудованої в певний програмний продукт, але також і означає обмеженість можливостей тонкого налаштування або оптимізації роботи СУБД.

PostgreSQL це дуже розвинена СУБД, яка в першу чергу орієнтується на реалізацію всіх стандартів SQL. Має одну із найкращих підтримок стандартів безпеки та багатопоточної обробки запитів. Реалізовує широкий спектр типів даних з можливістю створення власних типів, а також має можливість розширення функціоналу за допомогою плагінів (наприклад, *postgis* що забезпечує роботи із гео-даними). Незважаючи на велику кількість доступного функціоналу було вирішено, що використання даної СУБД буде надлишковим, більша частина можливостей та механізмів не потребується в нашому програмному продукті[10].

2.4.2 Вибір технологій розробки серверної частини

Основною мовою програмування було обрано Java, так як це багатопоточна, кросплатформенна, об'єктно-орієнтована мова програмування. Парадигма ООП дозволяє розробляти код, який легко розширювати та модифікувати, оскільки концепція об'єктів і взаємодій між ними запозичена із реального світу та інтуїтивно зрозуміла будь-кому[11]. Було обрано 8 версію Java, тому що на відміну від пізніших версій, вона розповсюджується за ліцензією «*Oracle Binary Code License*», яка дозволяє

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

вільно використовувати мову програмування, забороняючи лише її модифікацію та притягнення компанії Oracle до судових позовів пов'язаних з даною мовою програмування.

Для організації доступу до бази даних було використано ORM (Object-Relation Mapping - Об'єктно-реляційна проекція) фреймворк Hibernate. Після опису відповідності таблиць до об'єктів системи, програміст може зручно маніпулювати даними. Наприклад, виконувати CRUD (Create, read, update, delete) операції через програмні інтерфейси, що надає фреймворк. Перевагою такого методу є відсутність потреби написання SQL запитів, котрі місцями можуть бути доволі великим та складними у підтримці (наприклад з'єднання декількох таблиць в одному запиті). Також стандартний інтерфейс фреймворку забезпечує підтримку більшості типових запитів[12]. Наприклад, SELECT з умовами, обмежені за кількістю рядків вибірки, реалізація пагінації, тощо. Фреймворк поширюється за ліцензією «*LGPL-2.1*», яка дозволяє вільне використання програмного продукту і потребує лише вказання копірайту і тексту ліцензії в сирцевих кодах програми.

Для написання серверу було обрано фреймворк Spring[13], який надає широкий спектр засобів для реалізації різноманітного функціоналу. Фреймворк має відкритий програмний код, який поширюється за ліцензією Apache License 2.0. Було використано наступні компоненти:

- Spring Core, надає можливість створення IoC контейнера. Інверсія керування (англ. Inversion of Control, IoC) — це принцип побудови програми, при якому її частини отримують потік керування (викликаються) із загальної спільно використовуваної бібліотеки.
- Spring MVC, надає засоби для перехоплення та обробки HTTP запитів.
- Spring Security, надає можливості для реалізації захисту доступу до серверу.
- Spring Mail, надає зручний програмний інтерфейс для відправки листів через електронну пошту.

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

Окремо хотілось би розглянути функціонал роботи з поштовим сервісом, так як це по суті є центральним функціоналом програмного продукту. Комунікація із поштовим сервісом відбувається через протокол SMTP (Simple Mail Transfer Protocol, простий протокол пересилання пошти)[14]. В якості поштового сервісу було обрано поштовий сервіс від компанії Google Gmail. Вибір було зроблено у зв'язку із доступністю SMTP шлюзу для не комерційних поштових скриньок, що дозволяє отримати доступ до SMTP шлюзу одразу після реєстрації поштової скриньки.

Також було використано бібліотеку Flyway, яка забезпечує здійснення міграцій бази даних. Flyway надає механізми для контролю успішності проведених міграцій і відкату змін, в разі виникнення помилок в процесі міграції. Зокрема, у нашому випадку, не дивлячись на доволі просту структуру бази і невелику потребу у міграціях, бібліотека дозволяє значно спростити процес розгортання програмного продукту на нових машинах. Для кожної міграції створюється SQL файл з описом міграції і вказанням версії. Після цього для розгортання бази у СУБД, варто лише вказати дані для доступу до СУБД, такі як url, користувача та його пароль і описані міграції будуть виконані у відповідній СУБД при старті програмного продукту.

2.4.3 Вибір технологій розробки користувацького інтерфейсу

Angular це фреймворк від компанії Google для створення клієнтських додатків. Перш за все він націлений на розробку SPA-рішень (Single Page Application), тобто односторінкових додатків. Angular надає таку функціональність, як двостороннє зв'язування, що дозволяє динамічно змінювати дані в одному місці інтерфейсу при зміні даних моделі в іншому, шаблони, маршрутизація і так далі. Однією з ключових особливостей Angular є те, що він використовує в якості мови програмування TypeScript. TypeScript це мова програмування, що компілюється в JavaScript, але на відміну від нього надає програмісту можливість використання типів даних та класів із інтерфейсами[15].

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

Для спрощення розробки дизайну веб-сторінок було вирішено використати фреймворк Bootstrap, який надає цілий ряд стилів (мова йдеться про CSS стилі), що облегшують розробку адаптивного та лаконічного дизайну. Зокрема велику роль грає так звана «сітка Bootstrap». Вона дозволяє розбити сторінку на рядки, а рядки на колонки. Для різних розмірів екранів пристроїв можна вказати різне відношення розмірів елементів відносно один одного. Наприклад, на екрані комп'ютера або ноутбука показати додатково бокове меню, а на екрані смартфона відобразити текст на весь екран.

2.5 Розробка програмного продукту

2.5.1 Розробка серверної частини

Оскільки використовується бібліотека Flyway, немає потреби вручну створювати таблиці в базі даних, достатньо лише встановити СУБД і зробити опис таблиць в окремому SQL файлі в сирцевому коді програмного продукту та описати спосіб доступу до СУБД. Наприклад, так виглядає файл конфігурації доступу до СУБД написаний на мові розмітки yaml:

spring:

datasource:

driver-class-name: org.mysql.Driver

url: jdbc:mysql://localhost:5432/prog?currentSchema=message

username: prog

password: 1111

SQL-файли із міграціями мають спеціальні вимоги до найменування, для того щоб бібліотека могла їх правильно розпізнати, а саме такий шаблон: V1.0__migration_name.sql .

Сам файл міграції є нічим іншим як звичайним SQL-файлом:

V0.1__create_basic_tables.sql

```
CREATE TABLE usr
(
    id          serial PRIMARY KEY,
    email       varchar(200) NOT NULL UNIQUE,
    password    varchar(200) NOT NULL,
    name        varchar(100) NOT NULL
);

CREATE TABLE student
(
    id          serial PRIMARY KEY,
    owner_id    int          NOT NULL REFERENCES usr (id),
    name        varchar(200)  NOT NULL,
    email       varchar(200)  NOT NULL,
    student_id  varchar(20)   UNIQUE NOT NULL,
    group_name  varchar(10)   NOT NULL,
    debt        float4        NOT NULL
);

CREATE TABLE script
(
    id          serial PRIMARY KEY,
    usr_id      int          NOT NULL REFERENCES usr (id),
    name        varchar(100)  NOT NULL,
    text        varchar(1000) NOT NULL
);

CREATE TABLE scripts_criteria
(
    script_id   int          NOT NULL REFERENCES script (id),
    field_name  varchar(100) NOT NULL,
    condition   varchar(50)  NOT NULL,
    val         varchar(100) NOT NULL
);
```

Наступним кроком був опис відображення таблиць бази даних на програмні об'єкти.

```
@Entity
@Table(name = "usr")
public class User {
    @Id
    @Column(name = "id")
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    @Column(name = "name", nullable = false)
    private String name;
    @Column(name = "email", nullable = false)
    private String email;
    @Column(name = "password", nullable = false)
    private String password;
    @Column(name = "sign")
    private String sign;
    @OneToMany(mappedBy = "owner")
    private List<Script> scriptList;
    @OneToMany(mappedBy = "owner")
    private List<Student> studentList;
}
```

Анотація *@Entity* вказує на те, що даний клас містить опис відображення таблиці на об'єкт. За допомогою анотації *@Table* ми вказуємо таблицю з якої беруться дані. Анотація *@Id* вказує на те, що дане поле буде використовуватися, як ідентифікатор об'єкта. *@GeneratedValue* вказує на те, що значення буде згенеровано і визначає метод генерації, в нашому випадку значення буде згенеровано в базі даних та передано назад в систему. *@Column* описує назву колонки з якої потрібно взяти значення. *@OneToMany* позначає, що дане поле буде формуватися через зв'язок один-до-багатьох.

Далі описаний об'єкт можна використати при описі так званого репозиторія, інтерфейсу, який надає доступ до даних. Наприклад:

```
public interface UserRepository extends JpaRepository<User, Integer> {
    Optional<User> findByEmail(String email);
    Optional<Principal> findPrincipalByEmail(String email);
    boolean existsByEmail(String email);
}
```


Новий інтерфейс розширює інтерфейс JpaRepository, який в свою чергу описує типові методи роботи з таблицями, такі як findAll, findById, delete, save, тощо.

Далі було описано слухачі для різних HTTP методів, що забезпечує можливість взаємодії серверу з зовнішнім світом.

```
@RestController
@RequestMapping("/api/user")
@RequiredArgsConstructor
public class UserController {
    private final UserService userService;

    @GetMapping
    public Principal getPrincipal() {
        UserDetails user = (UserDetails)
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
        return userService.getPrincipal(user.getUsername());
    }

    @PutMapping
    public void updateUser(@RequestBody User user) {
        userService.updateUser(user);
    }

    @PutMapping("/{id}")
    public void changePassword(@PathVariable int id, @RequestParam
String password) {
        userService.changePassword(id, password);
    }

    @PostMapping
    public void createAccount(@RequestBody User user) {
        userService.createUser(user);
    }
}
```

@RestController вказує на те, що описані в даному класі обробники будуть надавати відповідь у форматі json.

@RequestMapping("/api/user") вказує на те, що посилання для описаних обробників буде починатися з /api/user.

@PostMapping, *@GetMapping*, *@PutMapping* вказують на те, що обробник прив'язується до POST, GET, PUT HTTP методів відповідно.

@RequestBody вказує, що тіло запиту буде конвертовано у вказаний клас.

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

2.5.2 Розробка користувацького інтерфейсу

Першим кроком при розробці користувацького інтерфейсу було визначення та опис сервіс-класів. Наприклад:

```
export class StudentService {
  constructor(private http: HttpClient, private securityService:
SecurityService) {}

  public findAll(): Observable<Student[]> {
    return
this.http.get<Student[]>(`/user/${this.securityService.getUserId()}/st
udent`);
  }

  public findById(studentId: number): Observable<Student> {
    return this.http.get<Student>(`/student/${studentId}`);
  }

  public removeById(studentId: number): Observable<void> {
    return this.http.delete<void>(`/student/${studentId}`);
  }

  public create(student: Student): Observable<void> {
    student.ownerId = this.securityService.getUserId();
    return this.http.post<void>('/student', student);
  }

  public update(student: Student): Observable<void> {
    student.ownerId = this.securityService.getUserId();
    return this.http.put<void>('/student', student);
  }
}
```

Основною ідеєю створення таких класів є створення обгортки, яка б представляла віддалений програмний інтерфейс нашого серверу у вигляді внутрішнього програмного інтерфейсу. Всередині відбуваються звичайні HTTP запити та десеріалізація отриманих відповідей у об'єкти.

Далі розробляються так звані компоненти – це об'єкти, якими оперує фреймворк при рендерингу сторінок. Компоненти можуть, як представляти цілі сторінки, так і описувати її певні елементи. Компонент складається із зв'язки CSS файлу, що описує стилі, які використовуються компонентом,

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

HTML файлу, який описує власне вигляд компоненту та TypeScript файлу, у якому описується логіка роботи компоненту (наприклад, які дані відобразити всередині компоненту і звідки їх взяти). Приклад компоненту який відповідає за модальне вікно підтвердження видалення об'єкту:

submit-modal.html

```
<div class="modal-header">
  <h4 class="modal-title">Ви впевнені?</h4>
</div>
<div class="modal-body">
  <p>{{description}}</p>
</div>
<div class="modal-footer">
  <button type="button" class="btn btn-success"
(click)="close(true)">Так</button>
  <button type="button" class="btn btn-danger"
(click)="close(false)">Ні</button>
</div>
```

submit-modal.ts

```
export class DangerousSubmitModalComponent implements OnInit {
  @Input() description: string;

  constructor(public activeModal: NgbActiveModal) { }

  ngOnInit() {
  }

  close(choice: boolean): void {
    this.activeModal.close(choice);
  }
}
```

Для того щоб створюваний клієнтський додаток зміг успішно функціонувати, залишилося лише описати внутрішню маршрутизацію додатку. А саме, відповідність внутрішніх посилань до компонентів які потрібно відобразити.

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

Приклад опису:

```
const routes: Routes = [  
  {path: 'login', component: LoginComponent, canActivate:  
    [NotAuthenticatedGuard]},  
  {path: 'registration', component: RegistrationComponent, canActivate:  
    [NotAuthenticatedGuard]},  
  {path: '', component: MainComponent, canActivate:  
    [AuthenticatedGuard]},  
];
```

Так при переході на відносне посилання /login буде відображено LoginComponent, який відповідає за вхід в систему.

					ДП 4671. 02.000 ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 2

1. По-етапне планування проєктованої системи, включаючи передбачувані технічні характеристики і функціональну складову, сприяє прискоренню роботи та покращенню якості фінального програмного продукту.
2. Обрані засоби реалізації програмного забезпечення, а саме: мова програмування, система управління базами даних, відповідні фреймворки та інші інструменти, найкраще виконують поставлені задачі та дозволяють розробляти якісні, легко масштабовані й ефективні системи.

					ДП 4671. 02.000 ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3

ТЕСТУВАННЯ ТА ОПИС ОТРИМАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

У процесі виконання дипломного проєкту вдалося створити вузьконаправлену інформаційну систему, яка значно автоматизує рутинну роботу викладачів по інформуванні груп іноземних студентів. А саме, була реалізована інформаційна система, яка допомагає викладачам одночасно відправляти велику кількість електронних листів, які стосуються навчального процесу і містять, насамперед, інформаційно-попереджувальний характер. Даний програмний продукт був створений, задля вирішення проблеми економії часу викладачів і оптимізації їхнього робочого процесу в цілому.

Уміння тверезо та раціонально оцінювати власні можливості та конкретно ставити перед собою необхідні задачі, дозволили у короткий термін реалізувати запланований функціонал на конкурентноспроможному рівні якості. Звичайно, не обійшлося і не без проблем, оскільки розробка такого роду інформаційних систем потребує відповідних знань, а особливо – досвіду.

Щоб переконатися у реалізації усіх запланованих функцій програмного продукту, розглянемо детальну інструкцію по експлуатації системи для користувача та проведемо відповідне тестування програмного забезпечення, щоб перевірити працездатність системи.

3.1. Інструкція користувача

Для початку роботи з системою необхідно створити новий аккаунт. Це можна зробити, натиснувши на кнопку «Створити обліковий запис!» у меню авторизації на веб-додаток, як показано на рис. 3.1.

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

Вітаємо в системі інформування ФІОТ!

Електронна пошта ❗

Пароль ❗

☐ Запам'ятати мене

Увійти

[Створити обліковий запис!](#)

Рис. 3.1. Вікно авторизації на веб-додаток

Далі, потрібно пройти звичну процедуру реєстрації, зазначивши ім'я користувача, назву його електронної скриньки та відповідний пароль. Усі поля даного вікна повинні обов'язково бути заповненими, про що свідчить знак оклику та червона рамка навколо них (рис. 3.2.)

Створити обліковий запис!

Ваше ім'я ❗

Електронна пошта ❗

Пароль ❗ Повторіть пароль

Створити

[Вже є обліковий запис? Увійти!](#)

Рис. 3.2. Вікно створення облікового запису

Після цього ми потрапляємо на основну робочу вкладку, так би мовити «головний екран» програми, з якого і будемо виконувати усі необхідні дії (рис. 3.3.)

Студенти		Скрипти		
#	Ім'я	Студентський	Група	Заборгованість
No data to display				
0 selected / 0 total				



Рис. 3.3. Головне вікно програми

Тепер, ми можемо працювати з системою. Розпочнемо з додавання до нашого списку необхідних студентів, яким і будуть адресуватися наші листи. Це можна зробити двома шляхами: вручну та за допомогою імпорту даних з excel-файлу.

Задля того, щоб додати нових студентів власноруч, потрібно натиснути на кнопку «плюс» у графічному інтерфейсі, яка виглядає, як зображено на рис. 3.4.



Рис. 3.4. Кнопка додавання нових студентів

Після цього, відкриється нове вікно, в якому ми можемо додавати студентів, заповнюючи усі поля, адже вони всі є необхідними для успішного виконання операції (рис. 3.5.)

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

Внесення нового студента

Ім'я

Назва групи

Номер студентського

Заборгованість

Електронна пошта

Зберегти

Відмінити

Рис. 3.5. Вікно додавання нових студентів

Ще одним способ додавання студентів є імпорт даних з ексель файлу. Цей спосіб є більш зручним і вимагає менших затрат по часу. Задля того, щоб ним скористатися, необхідно натиснути на кнопку, яка виглядає, як на рис. 3.6.



Рис. 3.6. Кнопка імпорту даних з excel-файлу

Після цього, відкриється вікно, в якому можна буде вибрати необхідний excel-файл (рис. 3.7.)

Імпортування списку студентів

Оберіть файл

Browse

Структура файлу

	A	B	C	D	E
1	Ім'я	Назва групи	Номер студентського	Заборгованість	Електронна пошта
2	James Bond	IK-62c	AR12312344	0	james@gmail.com

Завантажити шаблон

Зберегти

Рис. 3.7. Вікно імпортування списку студентів

Задля того, щоб вибрати необхідний файл, потрібно натиснути на поле «Оберіть файл» або кнопку «Browse», які є одним елементом (рис. 3.8.)



Рис. 3.8. Поле для вибору excel-файлу

Після цього, відкриється вікно провідника ОС, в даному випадку Windows 10, де можна вже привичним шляхом вибрати необхідний для імпорту excel-файл.

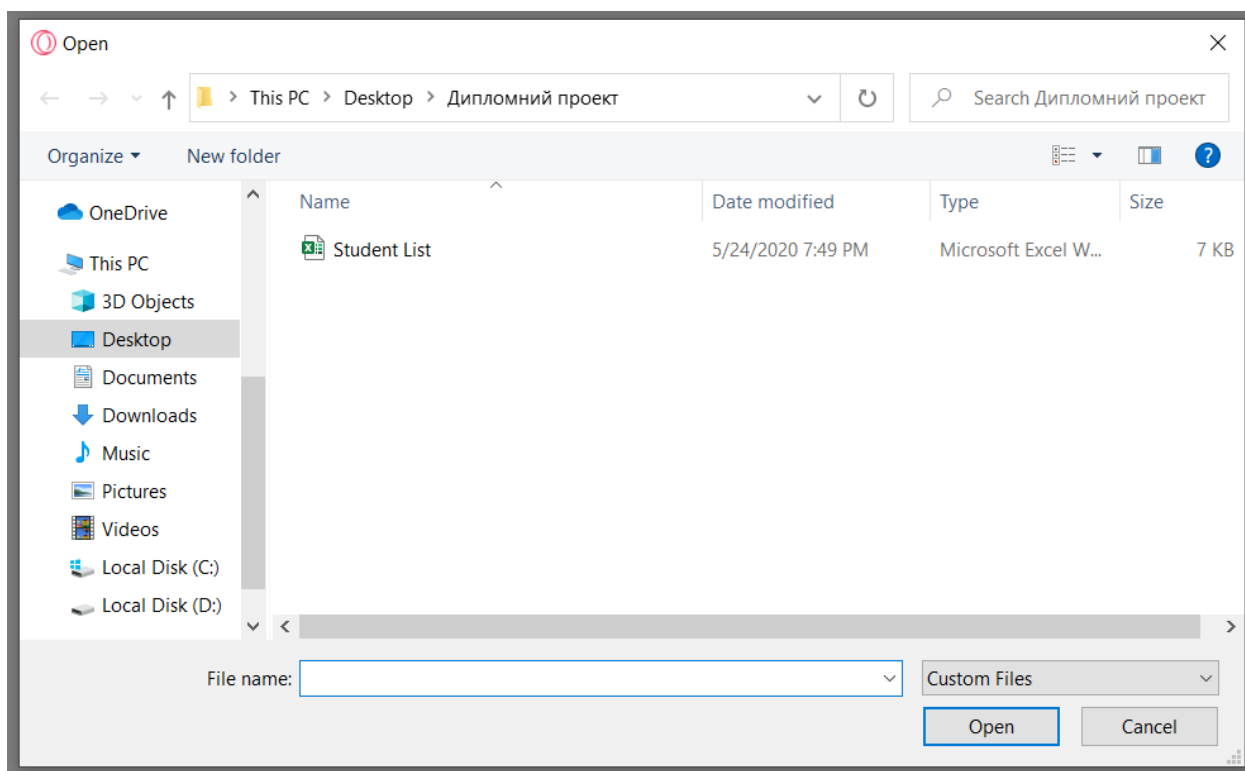


Рис. 3.9. Вікно провідника операційної системи

Також, у вікні імпорту можна завантажити готовий шаблон excel-файлу, задля того щоб не виникало зайвих проблем при імпорті даних. Для цього, потрібно натиснути на кнопку, яка виглядає, як показано на рис. 3.10.



Рис. 3.10. Кнопка завантаження excel-шаблону

Також, через головний екран веб-додатку можна відкрити вікно «Налаштування користувача». Для цього, потрібно натиснути на кнопку, яка виглядає, як зображено на рис. 3.11.



Рис. 3.11. Кнопка налаштування користувача

Після цього, відкриється вікно «Налаштування користувача», де можна змінити ім'я користувача, назву його електронної скриньки або пароль. Також, у цьому вікні можна змінювати універсальний підпис до усіх листів, які будуть відправлятися за допомогою цієї системи (рис. 3.12.)

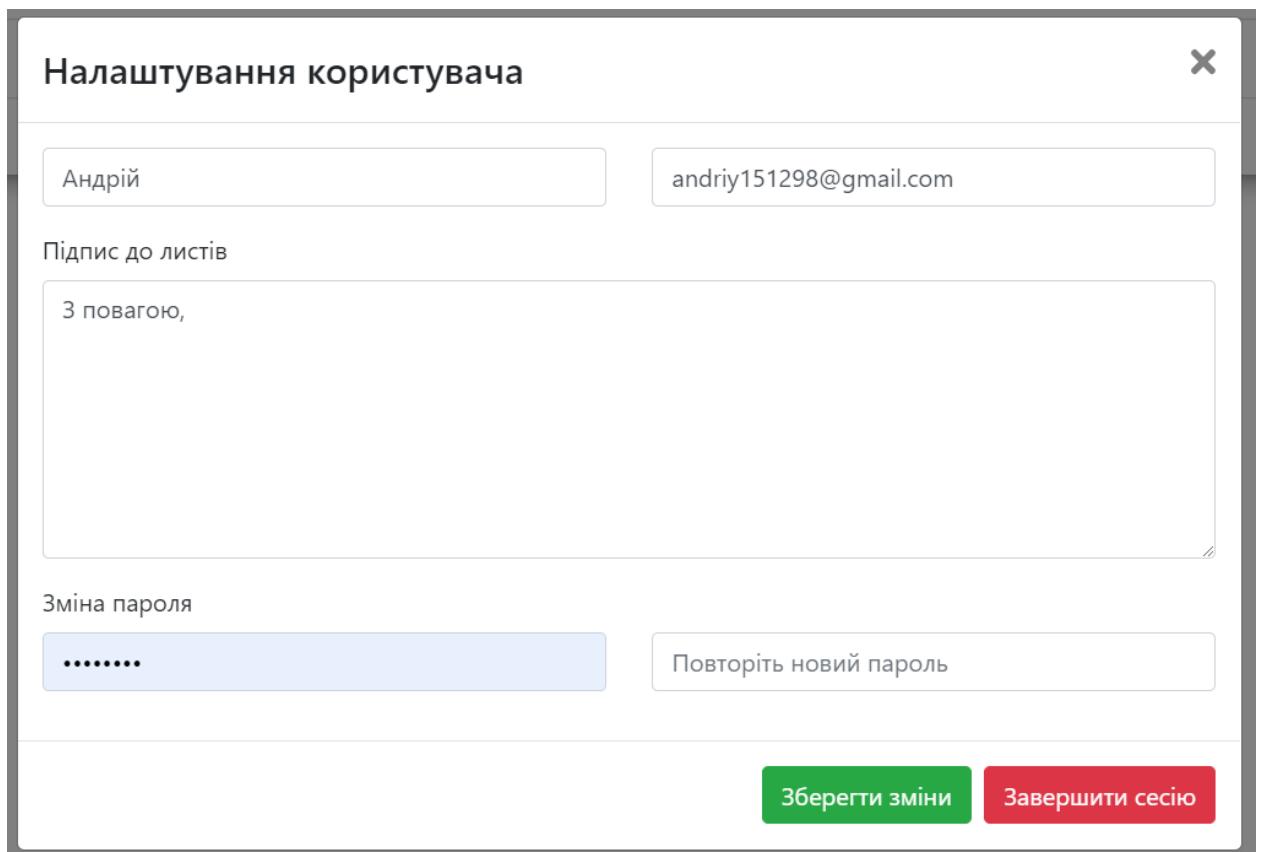


Рис. 3.12. Вікно налаштування користувача

Також, у вікні налаштування користувач може завершити поточну сесію та вийти зі свого облікового запису, натиснувши на кнопку, яка показана на рис. 3.13.

Завершити сесію

Рис. 3.13. Кнопка завершення сесії

Перейдемо до головного функціоналу системи, а саме відправлення електронних листів. Для цього, спочатку імпортуємо тестовий ексел-файл з готовими даними по студентах. Після цього, головний екран веб-додатку набуде вигляд, як зображено на рис. 3.14.

Студенти			Скрипти	
#	Ім'я	Студентський	Група	Заборгованість
<input type="checkbox"/> 1	Aaron Frye	KB43623523	ІК-61с	981
<input type="checkbox"/> 2	Adela Danson	KB67345345	ІК-61с	1765
<input checked="" type="checkbox"/> 3	Benedict Getson	KB66645345	ІК-61с	1405
<input type="checkbox"/> 4	Betty Kelly	KB12386567	ІК-61с	771
<input type="checkbox"/> 5	David Lawrenson	KB98704533	ІК-61с	202
<input type="checkbox"/> 6	Francis Jeal	KB23458348	ІК-61с	517
<input type="checkbox"/> 7	Smilla Nerger	KB00294245	ІК-62с	1608
<input type="checkbox"/> 8	Isaac Fisher	KB22556623	ІК-62с	1757
<input type="checkbox"/> 9	John Natt	KB21356346	ІК-62с	1830
<input type="checkbox"/> 10	Kory Mackall	KB23521435	ІК-62с	1984

0 selected / 25 total

Рис. 3.14. Головне вікно веб-додатку

Після цього, ми можемо відсортувати отримані поля по необхідному для нас параметру, будь то ім'я, група чи заборгованість. Задля цього, потрібно натиснути ЛКМ на назву потрібного нас стовпця. При цьому, при першому нажатті відбудеться сортування по зростанню значень, при повторному нажатті ЛКМ – по спаданню.

На рис. 3.15. зображено приклад відсортованого списку студентів по зростанню значення їхньої заборгованості на даний момент часу.

Студенти

Скрипти

#	Ім'я	Студентський	Група	Заборгованість ^	
<input type="checkbox"/>	1	Thad Rainford	KB23523567	IK-63c	0
<input type="checkbox"/>	2	Wilfred Shire	KB42347543	IK-63c	42
<input type="checkbox"/>	3	David Lawrenson	KB98704533	IK-61c	202
<input type="checkbox"/>	4	Rachel Yabsley	KB23412455	IK-63c	256
<input type="checkbox"/>	5	Rebecca Irwin	KB12632489	IK-63c	333
<input type="checkbox"/>	6	Nicole Hamill	KB23535690	IK-63c	340
<input type="checkbox"/>	7	Philip Morris	KB99076785	IK-63c	374
<input type="checkbox"/>	8	Zachary Comstock	KB16123055	IK-63c	408
<input type="checkbox"/>	9	Francis Jeal	KB23458348	IK-61c	517
<input type="checkbox"/>	10	Yolanda Blume	KB23572342	IK-63c	637

0 selected / 25 total

1

2

3

+

⚙

Рис. 3.15. Відсортований список студентів

Також, ми у будь-який момент можемо змінити дані щодо потрібного нам студента, просто натиснувши двічі ЛКМ на поле з його іменем. Після цього відкриється вікно редагування студента, яке зображене, як показано на рис. 3.16.

Редагування студента

Adela Danson

KB67345345

tgaracic@contcalpa.cf

IK-61c

1765

Зберегти

Видалити

Рис. 3.16. Вікно редагування студента

Тепер, щодо відправлення електронних листів. Це також можна зробити за допомогою двох методів: вибравши необхідних студентів зі списку

власноруч або ж створити автоматизовані сценарії, які при виконанні певних умов будуть самостійно розсилати електронні повідомлення.

Розпочнемо з першого методу – ручного. Задля того, щоб надіслати електронні листи необхідній групі студентів, спочатку потрібно вибрати їх зі всього списку, натиснувши на відповідні позначки поруч з їхнім ім'ям, як це показано на рис. 3.17.

Студенти					Скрипти				
#	Ім'я	Студентський	Група	Заборгованість					
✓ 1	Aaron Frye	KB43623523	IK-61c	981					
✓ 2	Adela Danson	KB67345345	IK-61c	1765					
✓ 3	Benedict Getson	KB66645345	IK-61c	1405					
✓ 4	Betty Kelly	KB12386567	IK-61c	771					
✓ 5	David Lawrenson	KB98704533	IK-61c	202					
✓ 6	Francis Jeal	KB23458348	IK-61c	517					
✓ 7	Smilla Nerger	KB00294245	IK-62c	1608					
✓ 8	Isaac Fisher	KB22556623	IK-62c	1757					
✓ 9	John Natt	KB21356346	IK-62c	1830					
✓ 10	Kory Mackall	KB23521435	IK-62c	1984					

10 selected / 25 total

1 2 3

Рис. 3.17. Вибір необхідних студентів

Після того, як ми вибрали принаймі одного студента, у нас з'являється кнопка для відправлення електронного листа, яка показана на рис. 3.18.



Рис. 3.18. Кнопка відправлення електронного повідомлення

Натиснувши на неї, відкриється вікно відправлення електронного листа, де користувач може ввести заголовок повідомлення та, безпосередньо, написати основний його зміст (рис. 3.19.)

Відправлення листа

×

Заголовок повідомлення

Текст повідомлення

?

name

studentId

groupName

debt

З повагою, Якубовський Андрій

Відправити

Відмінити

Рис. 3.19. Вікно відправлення листа

Звертаємо вашу увагу на те, що як вже було сказано вище, надпис: «З повагою, Якубовський Андрій» був створений автоматично і тому, не потрібно кожного разу дописувати його у кінець повідомлення.

Також, при формуванні основного тексту повідомлення, ми можемо користуватися загальними назвами необхідних параметрів, які при використанні у листі будуть замінюватися на відповідне значення параметра студента.

?

name

studentId

groupName

debt

Рис. 3.20. Загальні назви параметрів

- ? – зображує правила підстановки необхідного параметру
- name – ім'я студента
- studentId – студентський квиток
- groupName – назва групи
- debt – величина боргу

У нашій системі, задля того, щоб використовувати дані загальні параметри, необхідно з двох сторін оточити їх фігурними дужками, як це показано на рис. 3.21.

Текст повідомлення

? debt groupName name studentid

Доброго дня, {name}. Ваш борг на даний момент часу сягає {debt}. Просимо вас негайно оплатити необхідну суму, заради уникнення зайвих проблем.
З повагою, Якубовський Андрій

Рис. 3.21. Приклад використання загальних параметрів.

І тепер, ми можемо відправляти електронний лист, натиснувши на відповідну кнопку, яка показана на рис. 3.22.

Відправити

Рис. 3.22. Кнопка відправлення листа

Ще одним способом відправлення електронних повідомлень, являється використання готових скриптів. Задля того, щоб переглянути існуючі або створити новий скрипт, необхідно спочатку перейти у відповідну вкладку «Скрипти» (рис. 3.23.)

Студенти			Скрипти	
#	Назва	Тип повторення	Час останнього запуску	Увімкнений
No data to display				
0 total				



Рис. 3.23. Вікно перегляду скриптів

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

Щоб створити новий скрипт, потрібно ж знову ж таки скористатися вже звичною для нас кнопкою, яка зображена на рис. 3.24.



Рис. 3.24. Кнопка створення нових скриптів

Після цього, відкриється вікно створення нового скрипта, де можна вказати його назву, періодичність повторення, список необхідних студентів для розсилки, заголовок і текст повідомлення (рис. 3.25.)

Створення нового скрипта

Назва скрипту

Назва

Періодичність повторення

Параметри вибірки

Додати параметр

Попередній перегляд

Приклад вибірки

No data to display

Параметри повідомлення

Заголовок повідомлення

Текст повідомлення

? debt groupName name studentId

З повагою, Якубовський Андрій

Зберегти

Відмінити

Рис. 3.25. Вікно створення нового скрипта

Пройдемося детальніше по кожному з нових параметрів. При нажатті на поле періодичність повторення, користувачу видадуть випадаючий список, в якому він зможе вибрати період надсилання електронних листів (рис. 3.26.)

Періодичність повторення

Рис. 3.26. Випадаючий список можливих варіантів повторення

Також, у вікні створення скрипта користувач має змогу додати необхідні йому параметри, по яким будуть відбиратися студенти для розсилки листів. Задля цього, потрібно натиснути на кнопку, яка зображена на рис. 3.27.

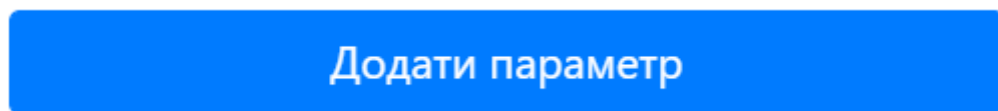


Рис. 3.27. Кнопка додати параметр

Після цього, стануть доступними нові поля, у яких можна буде вказати потрібні нам параметри вибірки, як показано на рис. 3.28.

Параметри вибірки

Рис. 3.28. Приклад параметрів вибірки

Також, можна у будь-який момент видалити необхідний параметр, натиснувши на кнопку, яка зображена на рис. 3.29.



Рис. 3.29. Кнопка видалення параметру

Для перегляду списку студентів, які попадають під задані параметри, можна скористатися кнопкою «Попередній перегляд» (рис.3.30.) і отримати список студентів, безпосередньо, для розсилки листів (рис. 3.31.)



Рис. 3.30. Кнопка попереднього перегляду

Приклад вибірки

Smilla Nerger	KB00294245	IK-62c	1608
Isaac Fisher	KB22556623	IK-62c	1757
John Natt	KB21356346	IK-62c	1830
Kory Mackall	KB23521435	IK-62c	1984
Émilie Ménard	KB74578346	IK-62c	1514

Рис. 3.31. Приклад вибірки

З параметрами повідомлення, а саме його заголовком і текстом ми вже розібралися вище. А задля того, щоб зберегти новий скрипт, потрібно натиснути на відповідну кнопку, яка зображена на рис. 3.32



Рис. 3.32 Кнопка збереження скрипта

Після цього, ми бачимо, що у вкладці скриптів добавився новий сценарій, який ми щойно створили (рис. 3.33.)

Студенти			Скрипти	
#	Назва	Тип повторення	Час останнього запуску	Увімкнений
▶ 1	Боржники	Щороку	-	<input checked="" type="checkbox"/>
1 total				

Рис. 3.33. Приклад нового скрипта

Як бачимо з рис. 3.33., ми можемо спостерігати назву скрипта, період його повторення, час останнього запуску та актуальний статус.

У будь-який момент часу, можна власноруч запустити необхідний для нас скрипт. Для цього потрібно натиснути на кнопку, як показано на рис. 3.34.



Рис. 3.34. Кнопка ручного запуску скриптів

Також, на даній вкладці можна активувувати/деактивувувати створені скрипти, використовуючи перемикач, який зображений на рис. 3.35.



Рис. 3.35. Перемикач активності скриптів

Це весь опис доступного функціоналу, яким ви можете скористатися, використовуючи нашу інформаційну систему.

3.2. Моделювання роботи системи

Задля того, щоб перевірити працездатність системи, відправимо електронні листи трьом студентам. Скористаємося ручним відправленням, вибравши необхідних студентів.

На рис. 3.36. ми можемо спостерігати шаблон, який буде відправлено усім трьом студентам, яких ми обрали випадковим чином.

Відправлення листа

Отримання студентських квитків

Текст повідомлення

?

name

studentId

groupName

debt

Доброго дня, {name}. Студентські квитки вашої групи {groupName} уже готові. Ви можете забрати їх у період з 10:00 до 16:00 протягом цього тижня. Не забудьте звірити номер вашого студентського квитка. Ваш номер: {studentId}.
З повагою, Якубовський Андрій.

Відправити

Відмінити

Рис. 3.36. Шаблон листа

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

В результаті, кожен обраний нами студент отримав лист з наступним змістом, який можна переглянути на рис. 3.37., рис. 3.38., рис. 3.39.

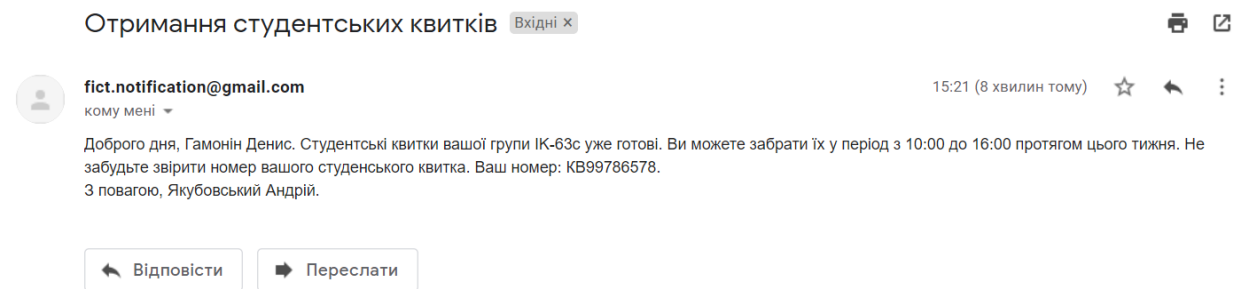


Рис. 3.37. Зміст першого листа

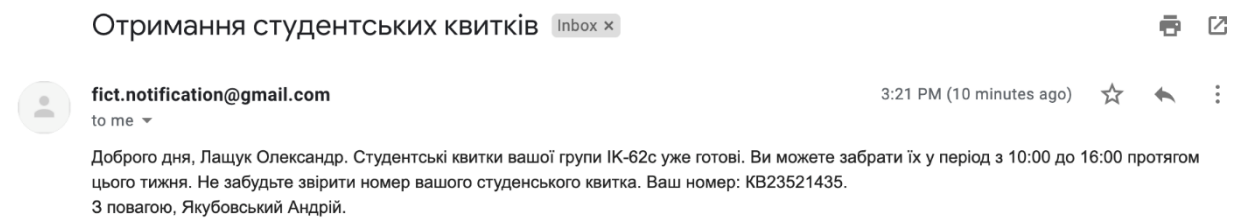


Рис. 3.38. Зміст другого листа

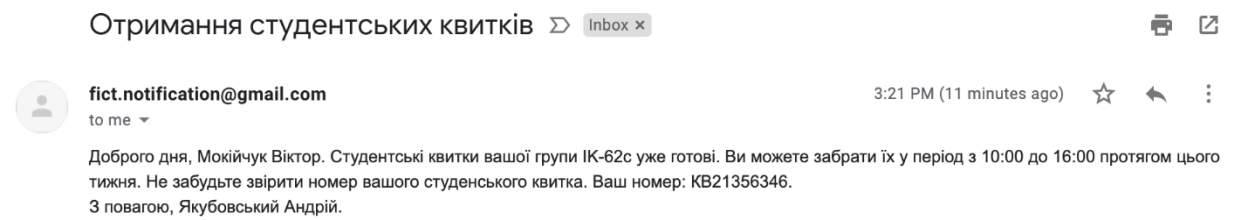


Рис. 3.39. Зміст третього листа

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Було розглянуто весь доступний функціонал системи та створена детальна інструкція користувача, що сприяє виникненню більш дружньої, зрозумілої та зручної взаємодії клієнта з системою.
2. Було проведено тестове моделювання системи, яке повністю підтверджує її працездатність і готовність до використання кінцевим користувачем.

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

ВИСНОВКИ

Даний дипломний проєкт являє собою комплекс дій, у який входять аналіз предметної області, постановка існуючих проблем і шляхи їх вирішення, аналіз уже існуючих програмних рішень і розробка нового конкурентноспроможного продукту, який призначений задля економії часу викладачів.

У ході роботи було досліджено чималу кількість літератури та різноманітних електронних джерел. Була створена порівняльна характеристика, яка включає в себе переваги та недоліки схожих за функціоналом інформаційних систем.

Також, було детально обґрунтовано вибір мови програмування, системи управління базами даних і інших інструментів, які використовувалися задля розробки програмного забезпечення.

У дипломному проєкті був наданий детальний опис всієї системи разом з усіма тонкощами, які необхідно знати задля зручної та ефективної роботи з нею, а проведенне тестування відправки листів декільком студентам лише доказує її працездатність.

Основною метою створення даної інформаційної системи є оптимізація робочого процесу керівників, які відповідають за групи іноземних студентів, шляхом налагодження автоматизованої відправки електронних листів. І таким чином, викладачам не потрібно займатися рутинною роботою та витратити дорогоцінний час на самостійне надсилання великої кількості електронних повідомлень. І також, система покликана допомогти іноземним студентам у якісному, а головне своєчасному інформуванні зі сторони навчального закладу.

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

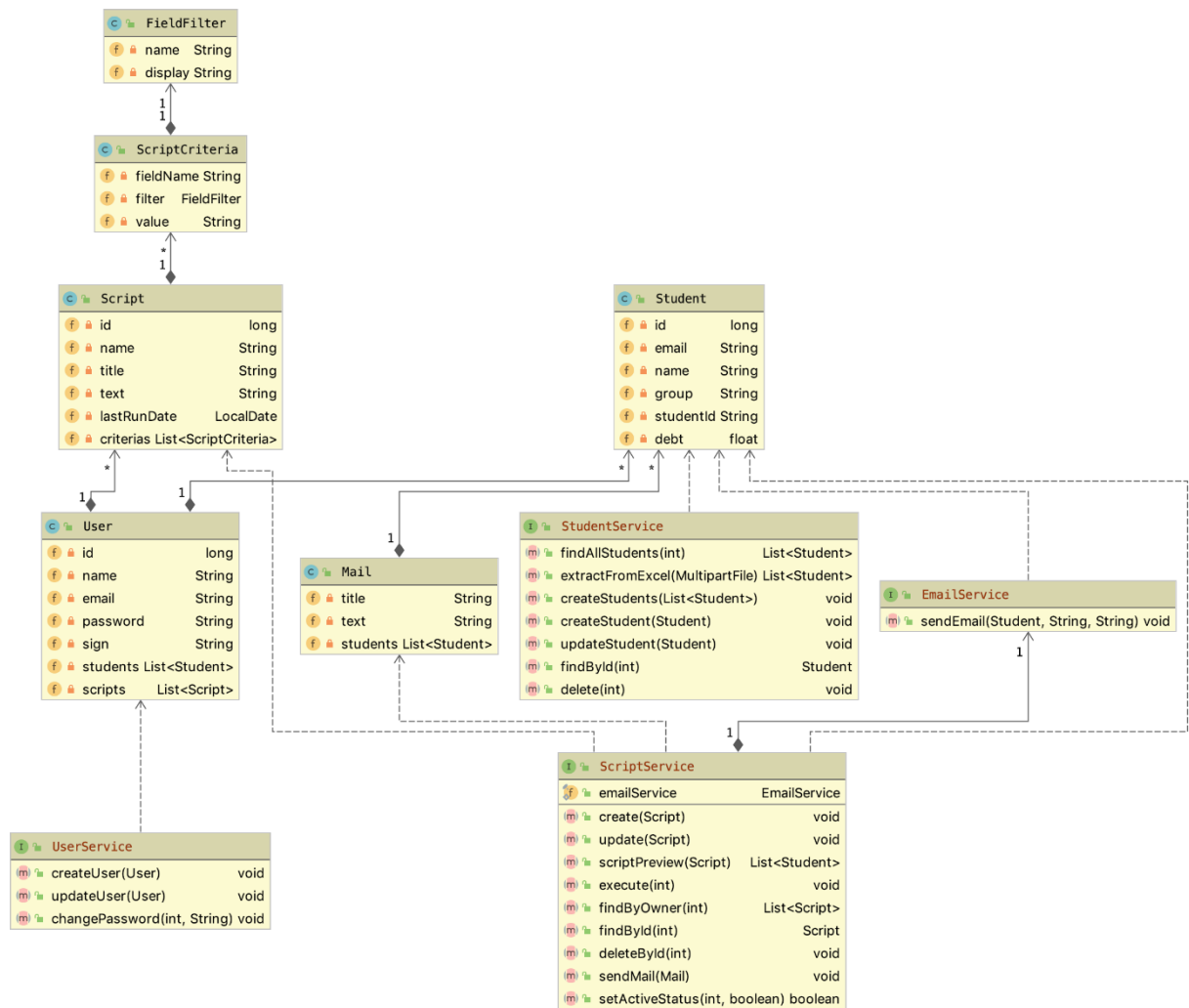
1. Інформаційні системи та технології: навч. посіб. для студентів за напрямом підготовки «Транспортні технології» / О. В. Грицунов; Харк. нац. акад. міськ. госп-ва. – Х.: ХНАМГ, 2010. – 222 с.
2. Информационные системы : учеб. пособие / Е.В. Бурцева, И.П. Рак, А.В. Селезнев, А.В. Терехов, В.Н. Чернышов. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2009. – 128 с.
3. Know the features and functions of information systems [Електронний ресурс]. – SlideShare, 2018. – Режим доступу: <https://www.slideshare.net/RajeshKhadka1/know-the-features-and-functions-of-information-systems> – Дата звернення : 16.03.2020.
4. Информационные системы: учебник для студ. учреждений сред. проф. образования / Г. Н. Федорова. — 3-е изд., стер. — М. : Издательский центр «Академия», 2013. — 208 с.
5. Информационные системы / Петров В. Н. — СПб.: Питер, 2003. — 688 с.
6. Автоматизированные информационные системы : учебник для студ. учреждений сред. проф. образования / К.Н.Мезенцев. — 4-е изд., стер. — М. : Издательский центр «Академия», 2013. — 176 с.
7. Автоматизация информационных процессов [Електронний ресурс]. – Материалы по информационным технологиям, 2009. – Режим доступу: <http://inftis.narod.ru/is/is-n3.htm> – Дата звернення : 19.03.2020.
8. Самоучитель UML / Александр Леоненков — 2-е изд. — Издательство «БХВ-Петербург», 2004. — 418с.
9. METANIT.COM Сайт о программировании [Електронний ресурс]. – Руководство по проектированию реляционных баз данных, 2009. – Режим доступу: <https://metanit.com/sql/tutorial/1.3.php> – Дата звернення : 23.03.2020.

					ДП 4671. 02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

10. Tproger Издание о разработке [Электронный ресурс]. – SQLite, MySQL и PostgreSQL: сравниваем популярные реляционные СУБД, 2016 – Режим доступа: <https://tproger.ru/translations/sqlite-mysql-postgresql-comparison/> – Дата звернення : 27.03.2020.
11. Java 8 in Action / Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft — Manning Publications, 2014 — 394с.
12. Hibernate Documentation – Режим доступа: <https://hibernate.org/orm/what-is-an-orm/> – Дата звернення : 06.04.2020.
13. Spring Boot in Action / Craig Walls – Manning Publications, 2015 – 264с.
14. RFC 5321 – Simple Mail Transfer Protocol / J. Klensin – IETF, 2008 – Режим доступа: <https://tools.ietf.org/html/rfc5321> – Дата звернення : 09.04.2020.
15. Angular документація – Режим доступа: <https://angular.io/docs> – Дата звернення : 11.04.2020.

Додаток 1
до дипломного проєкту
на тему: «Система автоматизованого інформування
іноземних студентів»

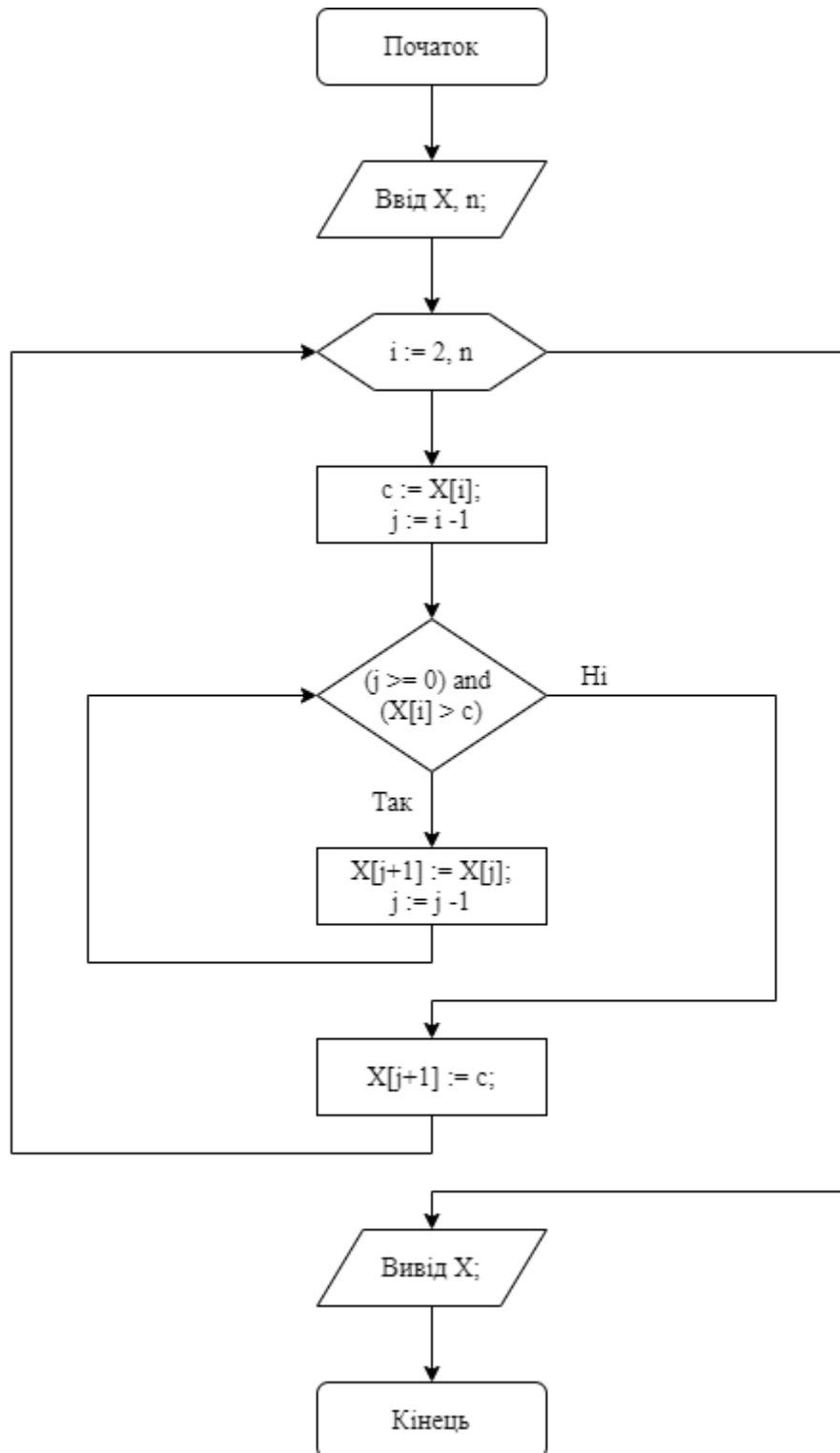
Київ – 2020 року



					ДП 4671. 03.000 Д1		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробник	Якубовський А.Ю.				Система автоматизованого інформування іноземних студентів Функціональна схема діаграми класів проекту	Лім.	Аркуш
Керівник	Павлов В.Г.						1
Н/контр.	Сімоненко В.П.					Аркушів	
Зав. каф.	Стіренко С.Г.					1	
						НТУУ "КПІ ім. Ігоря Сікорського", ФІОТ, ІО-62	

Додаток 2
до дипломного проєкту
на тему: «Система автоматизованого інформування
іноземних студентів»

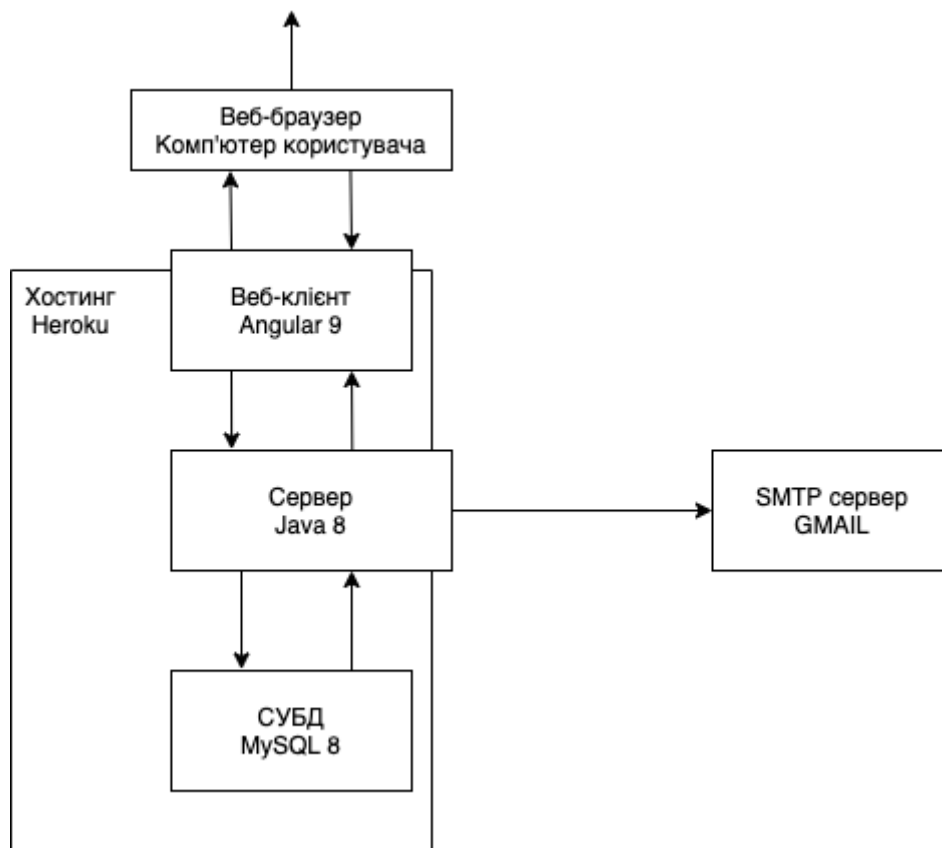
Київ – 2020 року



					ДП 4671. 04.000 Д2		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробник		Якубовський А.Ю.			Система автоматизованого інформування іноземних студентів Принципова схема алгоритму сортування вставками	Літ.	Аркуш
Керівник		Павлов В.Г.					1
						Аркушів	
						1	
Н/контр.		Сімоненко В.П.				НТУУ "КПІ ім. Ігоря Сікорського", ФІОТ, ІО-62	
Зав. каф.		Стіренко С.Г.					

Додаток 3
до дипломного проєкту
на тему: «Система автоматизованого інформування
іноземних студентів»

Київ – 2020 року



					ДП 4671. 05.000 ДЗ		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробник	Якубовський А.Ю.				Система автоматизованого інформування іноземних студентів Структурна схема системи	Літ.	Аркуш
Керівник	Павлов В.Г.						1
						Аркушів	
						1	
Н/контр.	Сімоненко В.П.					НТУУ "КПІ ім. Ігоря Сікорського", ФІОТ, ІО-62	
Зав. каф.	Стіренко С.Г.						